TEACHING COMPUTING WITH MAKEBLOCK MBOT

NICHOLAS HUGHES
& CARLOS IZSAK

Makeblock

TEACHING COMPUTING WITH MAKEBLOCK MBOT

NICHOLAS HUGHES
& CARLOS IZSAK

Book written by Nicholas Hughes and Carlos Izsak for Makeblock Co., Ltd.

Design by Kiko Núñez

First Published, Dec 2018
Makeblock Co., Ltd.

3rd Floor, Building C3, Nanshan iPark,
No.1001 Xueyuan Avenue, Nanshan District,
Shenzhen, Guangdong Province, China

https://www.makeblock.com

CONTENTS

Why use Robotics in the classroom?

1 INTRODUCTION

This book is aimed at educators interested in teaching computing using robots, in particular but not exclusively Primary School teachers in the UK. We have chosen the Makeblock mBot as this is one of the best robots for the task: it is easy to use, is great value for money and you can achieve great things with it.

We structured the book to be easy to follow to allow you and your students build programming skills step by step with engaging lessons and fun challenges to try your newly acquired skills.

The content has been written with UK Primary National Curriculum in mind and it can also be used for teaching other curriculums or in more informal contexts such as coding or robotics clubs. It could even be used at home by parents!

We recommend using this book with children aged 8 to 11 years old who have some experience of Scratch programming language: mBlock the programming software use to program the mBot is based on Scratch. This book could also be used without any previous Scratch experience, in that case allow extra time per lesson as students will be learning coding with blocks and how to program a robot at the same time.

We've divided the book into 3 parts. The first part, Chapters 3 and 4, go through how to set up the mBot and using this book in more detail respectively.

The second part, Chapters 5 to 12, consists of seven Lessons (Chapters 6 to 12) while Chapter 12 introduces some of the more advanced mBot functions. All these chapters include UK National Curriculum links and are structured in three parts: introduction, main task and plenary with an overview of each Lesson in Chapter 5. The third and final part of the book (Chapters 14 to 22) consists of the Challenges: activities to test and have fun with your newly acquired robot skills.

In each of the Chapters we have included images, example codes and top tips, where appropriate, to help you and make it as easy as possible to follow. We have also included ideas how to extend each lesson and challenge. Once you and your students are done with all our lessons and challenges, you can come up with more challenges and activities of your own. We're sure your students will not be short of ideas! Remember that Makeblock robots are modular: you can add extra bits such as six legs, servos and light and sound packs or extra sensors which will open up a world of more fun, learning and new possibilities.

Welcome to world of robots and programming.

Enjoy the journey!

2 WHO ARE WE?

Nicholas Hughes

Nic is a self confessed IT geek. He is a class teacher, Head of Computing at Latymer Prep School, Primary CAS Master Teacher, CAS Hub leader, a Raspberry Pi Certified Educator, a Google certified Educator and works part-time as an educational consultant for 3BM.

He loves to explore the different ways technology can be used to enhance teaching. He is a massive fan of physical computing and robotics and how they can be used to support the teaching of programming.

In the past he worked alongside the education team in the London Borough of Redbridge as a AST for ICT, running training, supporting schools and writing curriculum. He has been lucky enough to speak at a number of conferences over the years about his work with robots, physical computing and the application of Games Based Learning in the classroom.

He has been exploring the mBot since it first appeared on Kickstarter as he was one of the original backers. He has been using them within his own school for over 3 years and runs training, though 3BM, acoss schools in London.

He is is eager for as many teachers as possible to learn more about Computing and how to teach it effectively.

Follow him **@duck_star** on twitter and he has a blog at **nicholashughes.blogspot.com.**

Carlos Izsak

Carlos has worked with Schools and Community Groups in and around London over the last 12 years. His work with Schools includes delivering outreach environmental, sustainability, science, history and geography projects with leading NGOs and Local Authorities. His community development experience includes setting up and supporting groups, building partnerships, organizing events and delivering training sessions. He has recently become fascinated with STEAM (Science, Technology, Engineering, Arts & Design and Mathematics) and Maker Education and has been building robots, 3d printers and learning electronics. Carlos is currently Maker Education Lead for Artefacto, developing Maker Education Services and The Makercart: a pop up makerspace for schools, libraries and community groups that includes a 3d printer, electronics, micro-controllers and robots complemented by books and resources. Most recent work includes helping libraries setting up makerspaces (Lab Central in Redbridge, Digital Dens in Kent and the new Oxford central library) by providing advice, equipment and staff training and also delivering workshops for the Institute of Imagination and Design Museum.

You can follow him on twitter **@Carlitosi71**

3 OUT OF THE BOX, GETTING THE MBOT SET UP TO THE CLASS

a. Choosing the right mBot

So you have chosen to go with mBots as your physical computing solution which version do you choose?

There are 2 types available to buy, Bluetooth and 2.4G. When you buy an mBot you have to choose one of the types. What you choose will depend on the hardware you have in school. The basic difference is how they communicate with the devices you have in school.

Hardware	Type of mBot
Tablets (iPad or Android)	Bluetooth
Mac	2.4G
Windows	2.4G
Chromebooks	Bluetooth

Don’t worry if your hardware changes in the future as you can upgrade the mBot to work with 2.4G or Bluetooth buy purchasing an extra part.

b. Assembly

So your boxes of robots have arrived, what’s next?

The mbots have to be built before you can actually use them in lessons.

Now it is up to you how you build them. You could build them each yourself, or get your class to do it. Children in Year 5 and 6 are quite capable of following the instructions in the box to assemble them. It will table about 45 minutes if they work in pairs.

You could build them as part of an initial lesson or get them built with a club or perhaps your more able students. Once they are built, I would advise that you keep them built. It makes things much easier when you want to use them again.

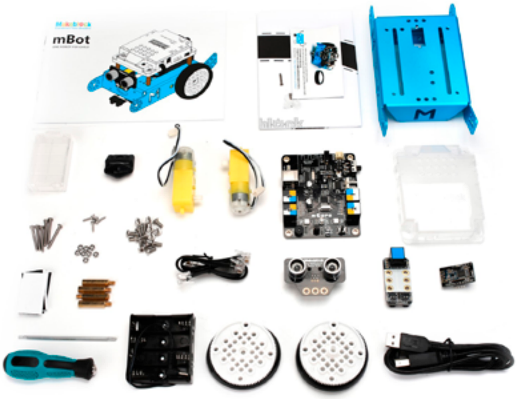


Figure 3. Parts included in the mBot box.

Some building tips:

- Make sure you check the motors are tightly secured before the wheels are attached. The bolts are difficult to get tight, you may need a small pair of pliers. They will need tightening from time to time. If you lose a bolt, don't worry you have plenty of spares in the kit.

- Don't worry too much about getting the motors plugged in the right way round, when you test it, if it is incorrect it is a quick adjustment. But the Left Motor plugs into M1 and the Right Motor plugs into M2.

- You may want to install the Bluetooth Module **(a)** or 2.4G Module **(b)** on the mCore **(c)** board yourself you are worried about the children bending the pins. It is the only delicate part of the build.

- It is important to plug the sensors into the correct ports on the mCore board. Ultrasonic **(d)** plugs into Port 3 and Line Follower **(e)** plugs into Port 2.

(a). Bluetooth Module

(b). 2.4 G Module

(c). mCore Board

(d). Ultrasonic Sensor

(e). Follower Sensor

- If you have the 2.4G version **make sure you label the Wireless Dongle (b) and the mBot. They are matched pairs, it is important not to mix them up.**

c. Testing

Now once they are build we need to test them.

You can control the mBot straight away using the included IR Remote Control. It functions like a TV remote so you have to point it in the direction of the robot and push a button.

Firsty make sure you have installed the batteries in both the mBot and the IR remote, they are not included. The mBot takes four AA batteries and the controller takes a CR2015 battery.

When you turn on the mBot it will make a few tones and the LED will be white. The mBot will be in IR Control Mode.

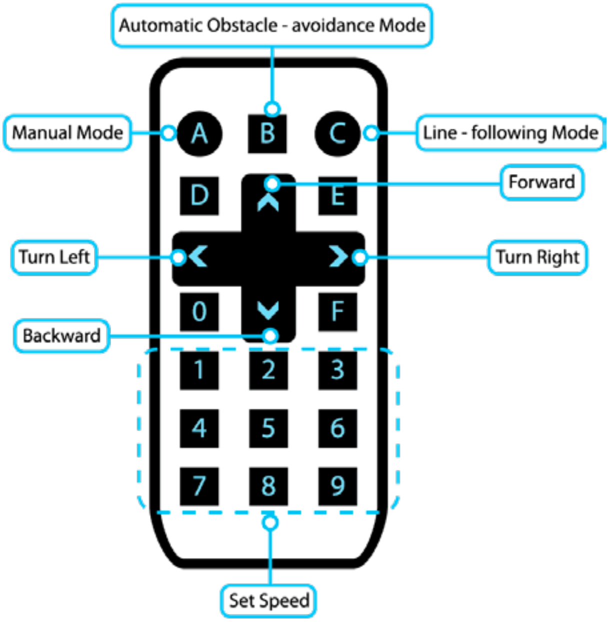


Figure 4. The IR controller: Once assembled you can use your mBot straight out of the box in three different modes: A-Manual model, B-Obstacle avoidance and C-Line following.

Try pushing some of the buttons. If when you push forwards and it moves backwards, you need to swap over the two motor wires which are plugged the mCore board **(c)**. Retest to make sure it moves as you expect. You can test the Sensors by pushing the B and C buttons.

If you push B, the LED will go green and will move around and avoid obstacles. If this doesn't work correctly check the Ultrasonic **(d)** is plugged into **Port 3.**

If you push C, the LED will go blue. Place your mBot on the line-follower **(e)** map and watch it follow the line. If this doesn't work correctly check the Line Follower plugged into **Port 2.**

Remember since the IR Remote Control is Infrared Red (IR) it can control more than one mBot at the same time. The controllers are not unique to each mBot, so they can be mixed up.

If you wanted to use the mBot with younger children using the IR Remote Control would be an ideal way to do it. The mBot will function just like a remote control car.

d. Installing software and firmware

Before you can start to program the mBot you will need to get the software installed. Please see table on the next page explaining what software needs to be installed for tablets (iPad or Android), Mac, PC or Chromebooks.

(b). 2.4 G Module

(d). Ultrasonic Sensor

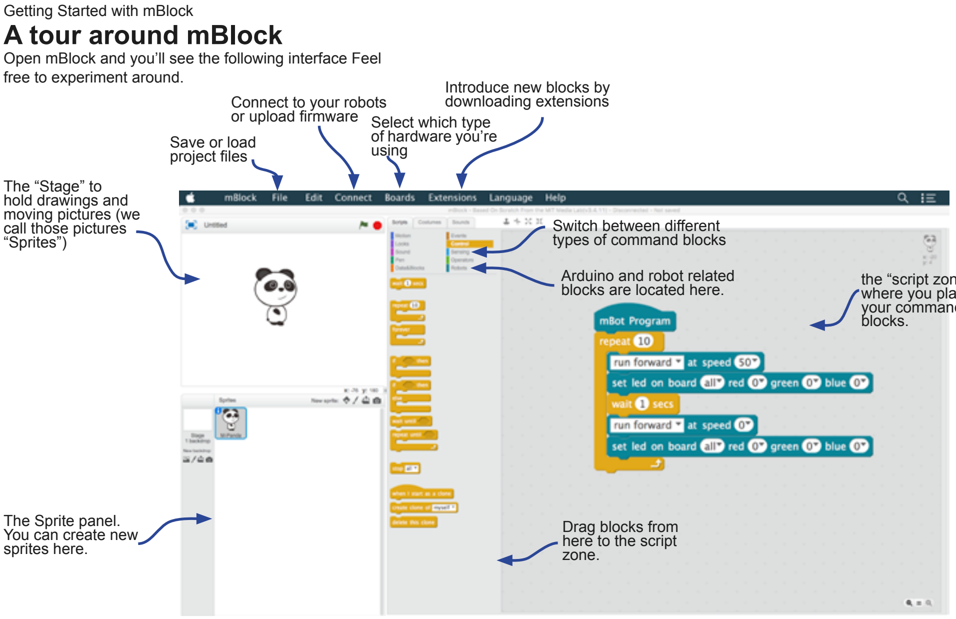
(c). mCore Board

Hardware	Type of mBot
Tablets (iPad or Android)	mBlock App from App and Google Play stores.
Mac	mBlock 3
Windows	mBlock 3
Chromebooks	mBlock App from chrome store

As well as choosing the software to be installed, it will need to be installed by your school's technician. See http://www.mblock.cc/software/ for more details and links.

At the time of this writing mBlock 5 is still in beta testing. This is based on Scratch 3, which will be fully released in Jan 2019.

The software that you install on Mac and Windows looks almost identical to MIT Scratch. If you have used scratch before you will notice a few small differences but it functions in exactly the same way.



Getting Started with mBlock

A tour around mBlock

Open mBlock and you'll see the following interface Feel free to experiment around.

Connect to your robots or upload firmware

Introduce new blocks by downloading extensions

Save or load project files

Select which type of hardware you're using

The "Stage" to hold drawings and moving pictures (we call those pictures "Sprites")

Switch between different types of command blocks

Arduino and robot related blocks are located here.

the "script zone" where you place your command blocks.

The Sprite panel. You can create new sprites here.

Drag blocks from here to the script zone.

```
mBot Program
repeat 10
    run forward at speed 50
    set led on board all red 0 green 0 blue 0
    wait 1 secs
    run forward at speed 0
    set led on board all red 0 green 0 blue 0
```

Figure 5. Image of mBlock computer interface

You drag blocks to the right of the screen and they can be snapped together like Lego bricks to create a program. The blocks relating to the mBot are under a new set of blocks called Robots.

The Stage is where you arrange sprites for something on screen to happen. If you want to learn more about how to use Scratch there are lots of resources on their own website. https://scratch.mit.edu/educators/

Now if you have bought the 2.4G Version of the mBot keep reading. If you have bought the Bluetooth Version of the mBot skip down to the next section.

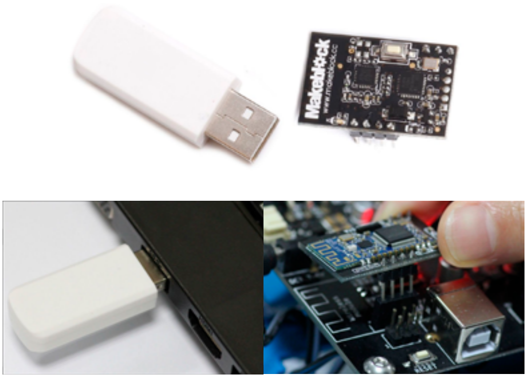


Figure 6. Image of 2.4G Wi-fi Dongle.

i. 2.4G Version

Most schools will choose to buy this version as it is the one recommended for education, assuming that you aren't using tablets or chromebooks. Make sure you label the Wireless Dongle **(b)** and the mBot. They are matched pairs, it is important not to mix them up.

In order to use the mBot with the programming software, mBlock, the firmware on the mBot needs to be updated. Doing this makes the IR Remote Control no longer work.

(b). 2.4 G Module

It is possible to do this in the initial lesson, but I would advise that it is done before the first lesson.

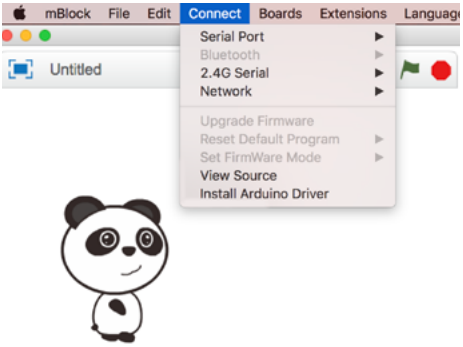


Figure 7. Image of menu in mBlock software

To update the Firmware firstly open the mBlock software. Then using the USB cable provided connect the mBot into the computer. The computer will detect, the first time you do this, that something has been plugged in. Let it install any drivers it needs to.

Then click on **Connect**, click on **Serial Port** and then click on **COMXX,** the number will vary. *If there is more than one option, unplug the mBot and see which Port vanishes. Plug it back in and the choose that one*.

Along the top it will now say **Connected**.

Now that the mBot is connected, go to **Connect** and click on **Upgrade Firmware**. A box will pop up and the mBot will then be upgraded.

If you ever want to reset the mBot to how it functioned out of the box, plug in the mBot again, connect to it using the **Serial Port** and then choose **Reset Default Program.**

Now when you turn on the mBot it will make one tone, this is telling you that it is set up to be used with the mBlock software.

Once the mBots are built and have had their Firmware upgrade, take all the remote controls, screwdrivers, USB cables and spares and store them somewhere safe. You won't need them for most lessons, but will at some point.

ii. Bluetooth Version

If you are using mBot with tablets or chromebooks, then you will need the Bluetooth version of the mBot.

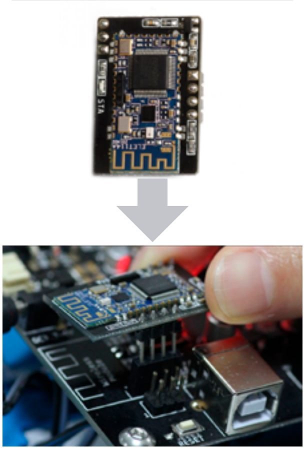


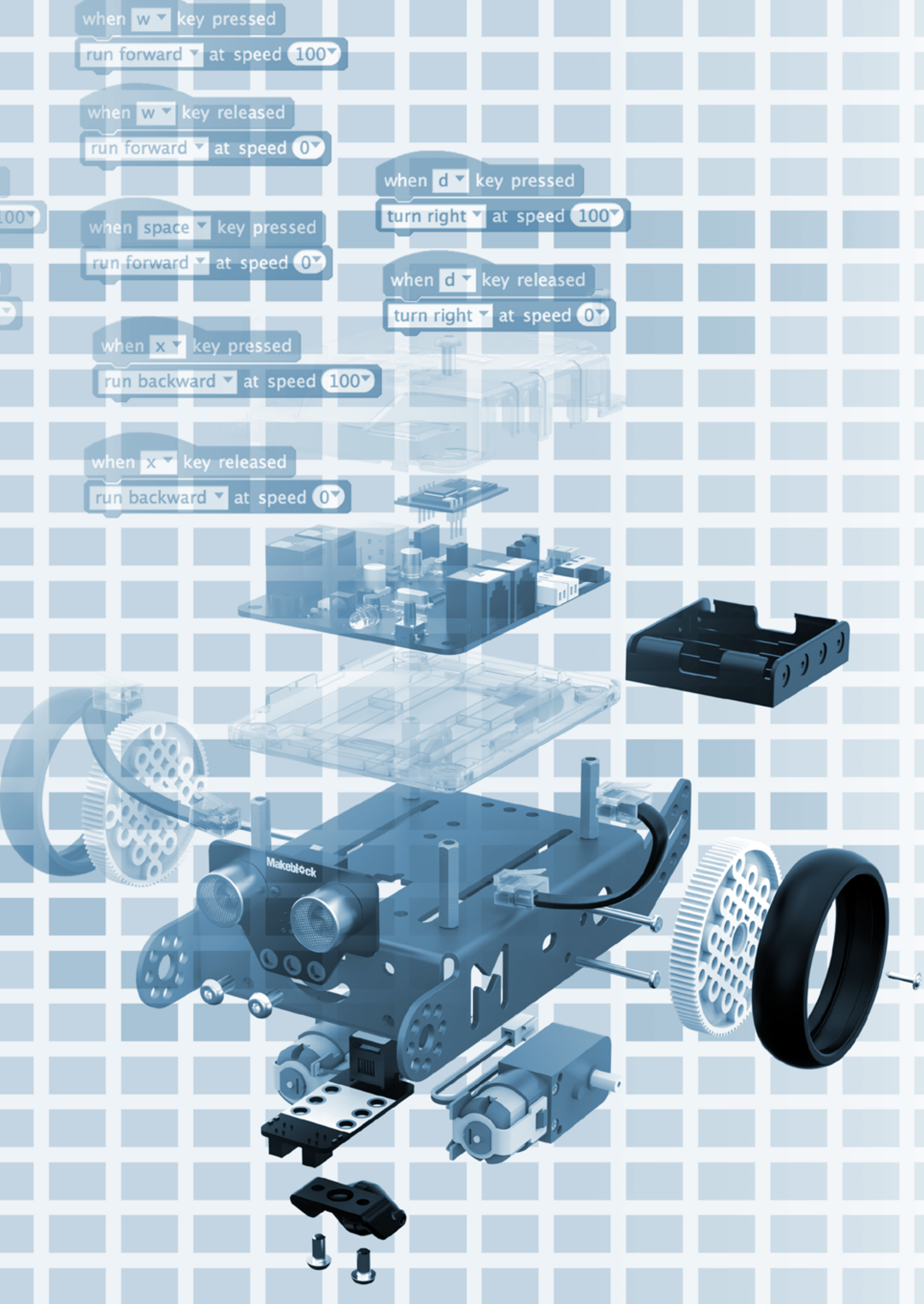
Figure 8. Image of Bluetooth Module Dongle

In order to program the mBot, all you to do is to connect the mBot via Bluetooth. No firmware needs upgrading. Within the app it will show you how to connect to the mBot. You just need to make sure that Bluetooth is turned on.

Once the mBots are built take all the remote controls, screwdrivers, USB cables and spares and store them somewhere safe. You won't need them for most lessons, but will at some point.

Top Tip:

When you turn on an mBot it makes a sound:
Dum, Dum, Dum = in starter mode.
Dum = in mBlock control mode
Take the wireless USB and label both it and the mBot, it is important that you keep them together as matched pairs



```
when w key pressed
run forward at speed 100

when w key released
run forward at speed 0

when space key pressed
run forward at speed 0

when x key pressed
run backward at speed 100

when x key released
run backward at speed 0

when d key pressed
turn right at speed 100

when d key released
turn right at speed 0
```

4 HOW TO USE THIS BOOK

The next part of the book will take you through a series of lessons to teach the basics functions of the mBot to children.

Later on it will showcase a series of challenges that you can use to further extend the learning, once the lessons have been taught. Each challenge explains what what lessons you should have completed first before attempting them. More on the challenges later in the book.

Each lesson will follow the following headings:

Heading	Explanation
Overview	What the lesson is about, highlighting the skills being taught.
NC Links	How it links to the English National Curriculum for Computing.
Lesson objective	What the children will learn in the lesson.

Then it will break the lesson into 3 parts and go into series of detailed step by step instructions.

	Explanation	**Approx. Timing**
Intro	Whole class input, explaining the lessons main concept which includes activities.	15 Minutes
Main Task	The activity that the children will be completing. Including differentiation to extend and support children in the class.	30 Minutes
Plenary	Reviewing their work and learning.	10 minutes

This book can be used as an introduction to block based programming, but I would advise that children have completed some work previously using block based programming on screen, ideally using Scratch. mBlock software is based on Scratch.

With all of these lessons it is very important for you to have worked though the lessons and activities before teaching them to the children. This will increase your confidence in teaching it and also help you troubleshoot any problems the children are having.

Each of the lessons are written to last around 45 minutes to an hour, this will vary from class to class. You can and should adapt the lessons covering less or more in each lesson as you see fit.

For most activities children should be working in small groups of 2 or 3, with one mBot per group.

5 OVERVIEW OF THE LESSONS

Lesson 1

Getting moving - getting the robot to move.

Overview: In this lesson we will be teaching them how to connect the mBot, introducing the mBlock interface to the children and getting the robot to move using the keyboard.

Lesson 2

Disco Time – understanding inputs and outputs.

Overview: In this lesson we will be learning about what inputs and outputs are and programming the mBot to dance and sing using the buzzer.

Lesson 3

Making me smile - further understanding of input and outputs.

Overview: In this lesson we will be learning how to add and use the LED matrix with the mBot. We will create an animated expression for the mBot.

Lesson 4

Sensing the world – an introduction to using an ultrasonic sensor.

Overview: In this lesson we will be learning how to use the ultrasonic sensor on the mBot. We will get the mBot to stop moving when it meets an object.

Lesson 5

Moving all by yourself - autonomous movement using an ultrasonic sensor.

Overview: In this lesson we will be developing our skills of using the ultrasonic sensor. We will get the mBot to move around by itself avoiding obstacles.

Lesson 6

Don’t fall – understanding a line follower sensor.

Overview: In this lesson we will be learning how to use the line following sensor. We will apply knowledge we have gained in previous lessons to make the mBot move around a table avoiding the edge.

Lesson 7

Remote control – how to reprogram the controller.

Overview: In this lesson we will learn how to program the mBot to respond to the IR remote control.

6

LESSON 1
GETTING MOVING - GETTING THE ROBOT TO MOVE

Overview:

In this lesson we will be teaching them how to connect the mBot, introducing the mBlock interface to the children and getting the robot to move using the keyboard.

NC Links

- 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

Lesson Objective

I can program the mBot to move using the keyboard.

Intro - 25 Minutes

(10 minutes - investigating precise instructions, 5 minutes - investigate mBot, 10 minutes - how to connect to mBot).

Pose question to the class. **What is a robot?** Discuss children's ideas. They may initially give answers like R2D2 and WALL-E. Challenge them to think about things that they see in the high street that are controlled by programing.

"A machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer."
(oxford dictionaries.com, 2017)

So a simpler definition is a machine that performs a task. Some examples include an ATM, Traffic lights and Automatic Tills.

Robot: *From Czech, from robota 'forced labour'. The term was coined in K. Čapek's play R.U.R 'Rossum's Universal Robots' (1920). (oxforddictionaries.com, 2017)*

Ask for a volunteer. Explain that they are going to pretend to be to a robot, they are going to perform a task, but have to follow the instructions the class gives them exactly.

You will need to have decided beforehand what task they will be performing considering what will work best in the space you have. You could simply ask them to move to a place in the classroom. You could ask them to move and pick something up such as a pen. The similar the better.

Once the class is clear on what the 'robot' will be doing, ask them to discuss in pairs what the instructions could be, they could note down what they think on whiteboards or paper.

For example if asking them to move to a place in the classroom children may come up with the following instructions, which would be imprecise.

Walk forward ← **For how long?**
Turn right ← **For how long?**
Walk forward
Turn left
Walk forward

The following instructions would be an example of more precise instructions.

Walk forward 5 steps. ← **Though what is a step? It must be the same each time.**
Turn right 90 degrees.
Walk forward 5 steps.
Turn left 90 degrees.
Walk forward 3 steps.

While the class are discussing this, talk to the volunteer and explain that they have to do only what they are told. If something doesn't make sense they should do nothing or say, 'Do not understand.'

For example if they are given the instructions turn left, they should just just keep turning left. It may be helpful, if you have another adult in the room, to ask them to be the robot first.

The class will give the instructions to the 'robot' and will have varied success. Work as a class and edit the instructions so they are more successful. **What was important to remember when giving instructions?** Agree that in order for the 'robot' to move correctly you have to give clear, unambiguous instructions.

Explain that you are going to program a robot called the mBot. Give a mBot to each pair/group. Ask the children to look carefully at the mBot. What do you notice? What sensors/capabilities do you think it has? Give them the Wireless Dongle as well so the mBot and Dongle stay together.

Discuss what they notice, no detail is too small. Discuss as needed what the parts are, sonic sensor, line follower, LEDs, motors etc...

Part	Function
mCore Case	Covers and protects the mCore
mCore	The 'brain' of the mBot. This is what allows allows the robot to be programmed.
Motor	Turns the wheels and it controlled by the mCore board.
Follower Sensor	A sensor that allows the mBot to recognise light and dark, so can follow a line.
Ultrasonic Sensor	A sensor that allows the mBot to sense if something is in front of it and how far away it is.
Mini Wheel	Allows the mBot to turn on the spot.

Figure 9. Image of mBot - labeled parts

mCore is unfortunately under a cover but the children may be able to pick out some details. You could try to open the cover so children can have a better look.

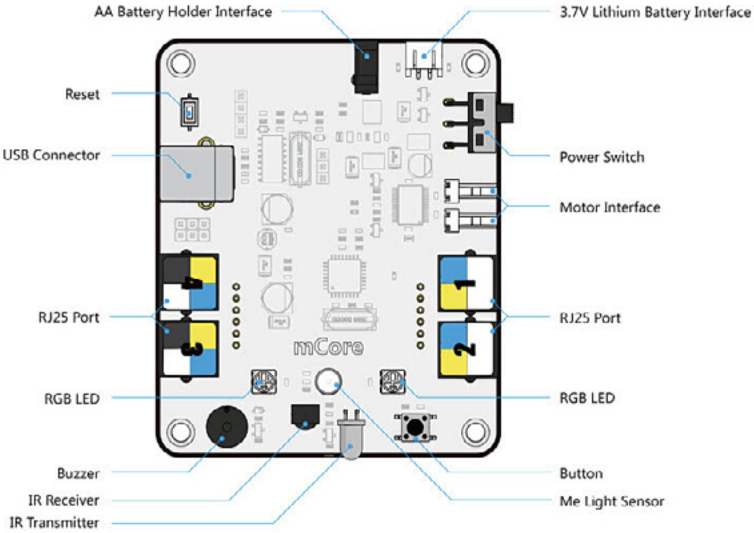
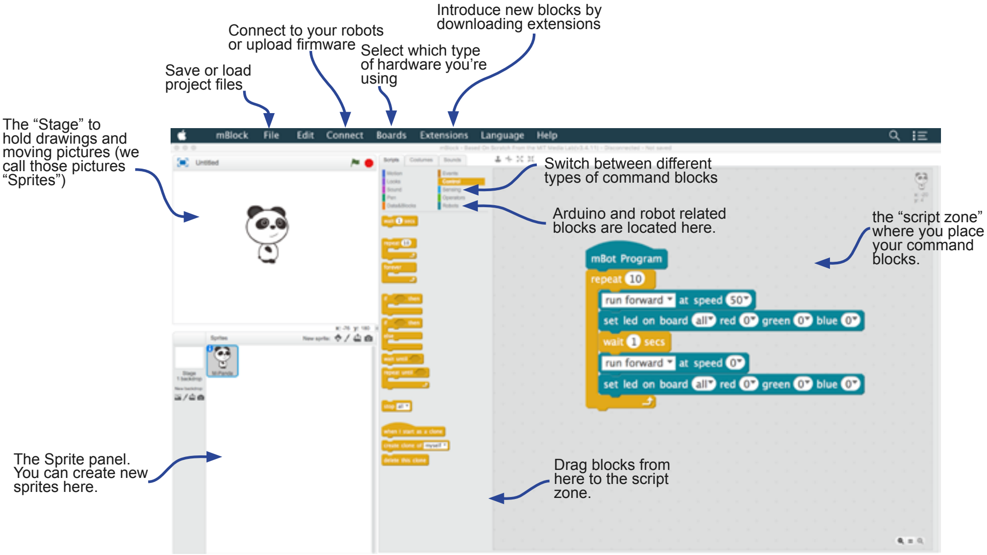


Figure 10. Image of mCore

Part	Function
RGB LED	A light that be programmed to shine different colours.
RJ45 Port	Allows sensors to be plugged in, they are uniquely numbered.

Explain that today we are going to program the robot to move.

Show mBlock programme, it may look familiar to children if they have done any work with Scratch. **What do they notice that is different?**



Connect to your robots or upload firmware

Introduce new blocks by downloading extensions

Select which type of hardware you're using

Save or load project files

The "Stage" to hold drawings and moving pictures (we call those pictures "Sprites")

Switch between different types of command blocks

Arduino and robot related blocks are located here.

the "script zone" where you place your command blocks.

The Sprite panel. You can create new sprites here.

Drag blocks from here to the script zone.

```
mBot Program
repeat 10
    run forward at speed 50
    set led on board all red 0 green 0 blue 0
    wait 1 secs
    run forward at speed 0
    set led on board all red 0 green 0 blue 0
```

Figure 11. Image of mBlock software

Review the main sections of mBlock.

Stage - where the sprites are placed, the Cat is now a Panda.

Blocks - where all the commands are, highlight the Robot section.

Script Zone - where you drag the blocks to and clip them together to create programs.

Demonstrate how to connect the mBot to the mBlock software.

Firstly the children will need turn on their mBot, **it should beep once.** If it doesn't you will need to look back to page 19 and check they have been set up correctly. There will be a flashing red light on the top of their mBot. Then the children will need to plug their Wireless Dongle into the USB port of their computer. Each mBot will have a paired Wireless Dongle, make sure they are not muddled up. The red light that was flashing should stop flashing when the Wireless Dongle has been plugged in. This shows that the mBot has been connected to the computer.

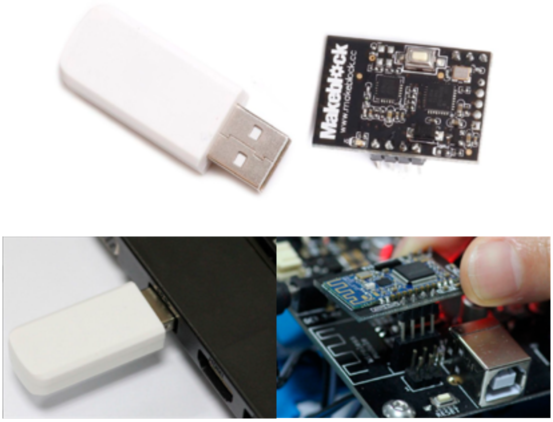


Figure 12. Image of wireless dongle.

You now need to connect the software to the mBot. Open up the mBlock software, go to Connect and click, then move down to 2.4G Serial, click on Connect. The heading at the top should change and say 2.4G Connected instead of Disconnected. The mBot is now connected to the computer ready to be programed.

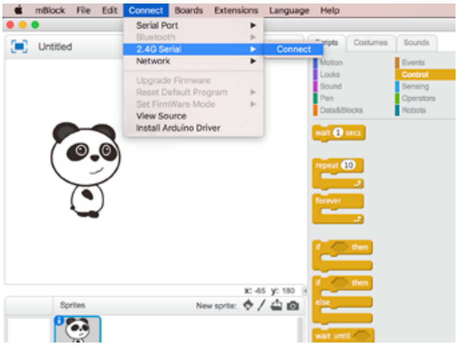
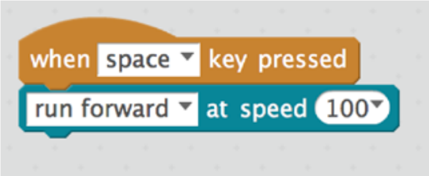


Figure 13. Image of mBlock Connect menu.

Main Task - 25 minutes

Show the class following code.



```
when space key pressed
run forward at speed 100
```

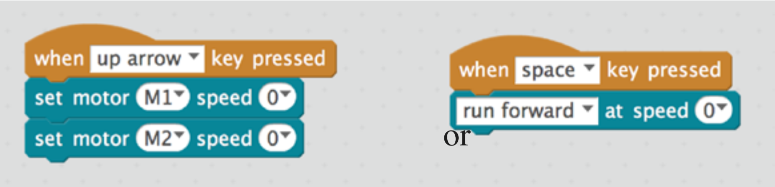
Figure 14. Image example code for moving forward

Ask them to discuss what they think it will do? Take children's responses. *Answer: The code makes the mBot move forward, it won't stop though.*

Ask children in pairs or small groups to use the dark orange and teal blocks to create this code and find out what it does for certain.

After a while get children to explain what happens and why doesn't it stop. Answer: *Because we haven't told it to.*

Ask children, **Can they work out how to make the mBot stop?** *You can show children the following code if needed. Most children will solve the stopping problem with the solution to the right. They will tell the mBot to stop when a key is pressed.*



```
when up arrow key pressed
set motor M1 speed 0
set motor M2 speed 0
```

or

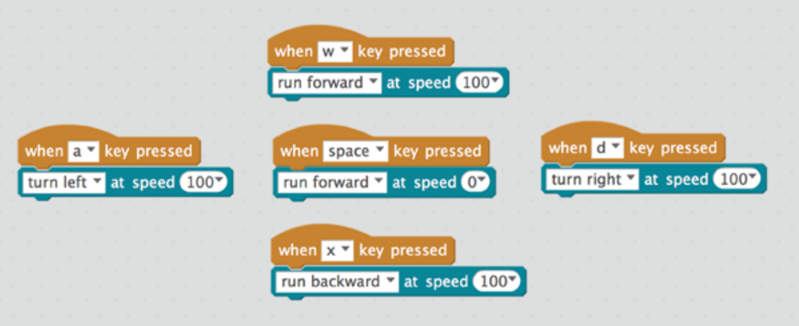
```
when space key pressed
run forward at speed 0
```

Figure 15. Image example codes for stopping

Set children a challenge. **Can you program the mBot to move around using keyboard inputs?** Children will need to consider what keys they are going to use, they need to make sure they have a way to stop the mBot moving.

Possible solutions:

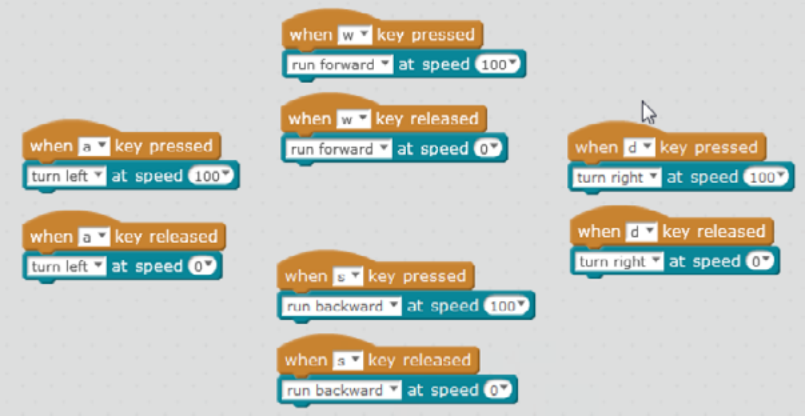
The following examples are similar, the first example (Figure 16) just uses key presses for events to happen.



```
when w key pressed
run forward at speed 100

when a key pressed
turn left at speed 100

when space key pressed
run forward at speed 0

when d key pressed
turn right at speed 100

when x key pressed
run backward at speed 100
```

Figure 16. Keyboard control example 1

The second example (Figure 17) is more advanced as it works more elegantly using keys being released to stop the mBot moving. Both are valid solutions.



```
when w key pressed
run forward at speed 100

when w key released
run forward at speed 0

when a key pressed
turn left at speed 100

when a key released
turn left at speed 0

when d key pressed
turn right at speed 100

when d key released
turn right at speed 0

when s key pressed
run backward at speed 100

when s key released
run backward at speed 0
```

Figure 17. Keyboard control example 2

Top Tips if something goes wrong.

- If for some reason the mBot disconnects from the computer, 2.4G Connected is no longer displayed along the top of the screen, get the children to save their work and then close the mBlock software. Then reopen it and reconnect to the mBot.
- If when you plug in Wireless Dongle USB the red light continues flash, you may have the wrong one. Remember label both the mBot and its paired Wireless Dongle and keep them together.

Extension

Make or mark out a simple track on the classroom floor. You could use tape or just some books. Get the groups to test out their controls. Are they accurate enough to get round the track? What changes could they make to make the controls more precise.

Differentiation

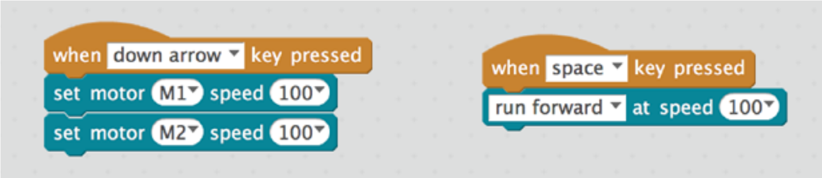
- Less able children could be given a pre-prepared file with some of the program already completed from them, for example the forward and backwards keys are already done. All they would need to work out is how to make it turn.

- More able children should be encouraged to explore what changing the value of the speed does. **What is the lowest value it can have and still allow it to move? What is the highest value that still an effect on the speed of the mBot?** *Much below 50 and the mBot will not move at all, the number refers to how much power is being sent to the motor. The highest number that has an effect is 255, you can put in higher numbers than that but it will go no faster.*

Plenary - 10 minutes

Gather groups together and review the challenge, did you program the mBot to move around using keyboard inputs? Ask some groups to share their solutions to the task. You can do this by moving round the class or displaying their work on a screen or interactive whiteboard. Ask groups to look at the programs being shared. **What do they notice that is similar and different?** *Answer: See examples above (Figures 16 and 17), the children will use a range of different keys and methods to stop,* note that they all use the when key pressed event block **Is there a best solution?** *Answer: Easy to understand, no extra unneeded blocks*. **Is there a solution that most of the class used?**

Show children the following two sets of blocks. What do they do?



```
when down arrow key pressed
set motor M1 speed 100
set motor M2 speed 100
```

```
when space key pressed
run forward at speed 100
```

Figure 18. Image example code for moving forward

Answer: The same thing. The mBot will move forwards.

Explain that there is always more than one way to solve a problem, **Which of these is better?** Depends what is important, the one on the left is easier to understand. But the one to the left will allow adjustment of the motors on each side so the mBot can travel in a straight line.

7

LESSON 2
DISCO TIME - UNDERSTANDING INPUTS AND OUTPUTS

Overview

In this lesson we will be learning about what inputs and outputs are and to program the mBot to dance and sing using the buzzer.

NC Links

- 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

Lesson Objective

I can program the mBot to dance.

Intro - 15 minutes

Pose a question to the class. **What is an input?** Children to discuss what they think. Take ideas from the children.

Explain that an input is when data is sent into a digital device. **Can you see anything on a computer that sends data inside it?** Children may mention keyboard, mice, webcam and microphone.

Pose another question to the class. **What is an output?** Children to discuss what they think. Take ideas from the children.

Output is data that a digital device sends out. **Again, can you see anything on a computer that sends data out?** Children may mention a screen, speakers/headphones and printers.

A hard drive/SD Card/Memory Card is neither a input or output, so that can be confusing, it is something that stores data.

Go this this site http://www.bbc.co.uk/guides/zx8hpv4 for a short video that you could watch and an interactive image that asks children to identify if a device is an input or output.

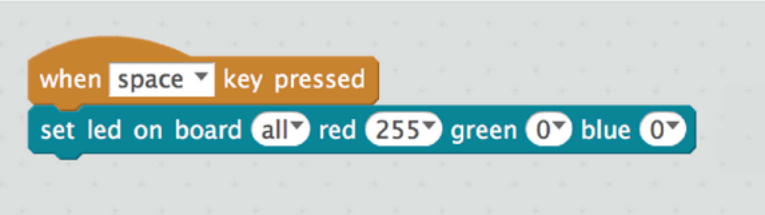
Give out the mBots to the pairs/groups of children. **On an mBot what are the input and outputs?** Give children some time to look back over the mBot and try and identify inputs and outputs. Get them to feedback what they think.

Input	**Output**
Children may not get all of these as some can't be seen through the mCore cover	
Follower Sensor	Motor
Ultrasonic Sensor	**On the board:** LEDs Buzzer IR Transmitter
On the board: Button IR Receiver Light Sensor	

Ask the class, **Where do the keyboard controls we used last week fit? Are they an input or output? Why?** *Answer: Input because data is being sent via the keyboard into the computer and then from the computer to the robot.*

Main Task - 30 minutes

Show children this code. What does this do?



```
when space key pressed
set led on board all red 255 green 0 blue 0
```

Figure 19. Image of example code

Answer: It makes the LEDs go red when the spacebar is pressed. Children to discuss and explain what they think.

How can we be sure? *Answer: Test out the code.*

Give each group/pair an mBot and paired Wireless Dongle. Ask children to make the code in mBlock and test their predictions. Children will need to connect mBot to mBlock software before they can test their predictions. See page 19 for instructions on how to connect the mBot to the mBlock software. **Where they correct?**

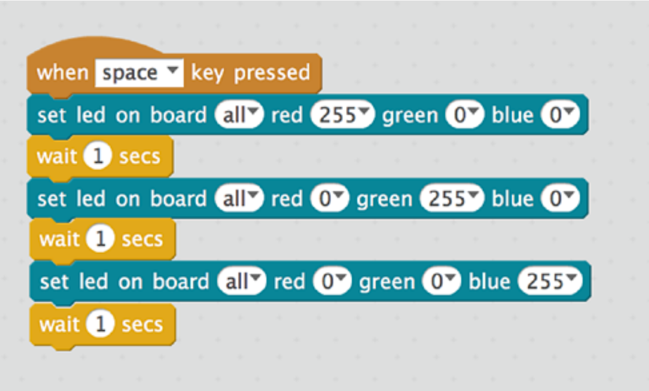
The code makes the LEDs on the mBot go red, ask children to see if they can get the mBot to show different colours. **How many different colours can they make?** They will need to change the values in red, green and blue. Move round the classroom encouraging children to keep making new colours. Give some hints as needed: **What happens if you have a different number in each space? Can you type in a number? What does changing 'all' do?**

RGB Values

The LEDs are coloured according to RGB values. Using a mix of the colours Red,not, Green and Blue other colours can be made. The values range from 0 to 255.

	(R, B, G)
Red	(255, 0, 0)
White	(255,255,255)
Purple	(128,0,128)
Yellow	(255,255,0)

Once all the children have made at least 4 different colours each, stop everyone and show the following code. **What does this do?** *Answer: When the spacebar is pressed the LEDs will go red, wait 1 second then change to green, wait for 1 second then change to blue*. Children to discuss and predict what the code will do.



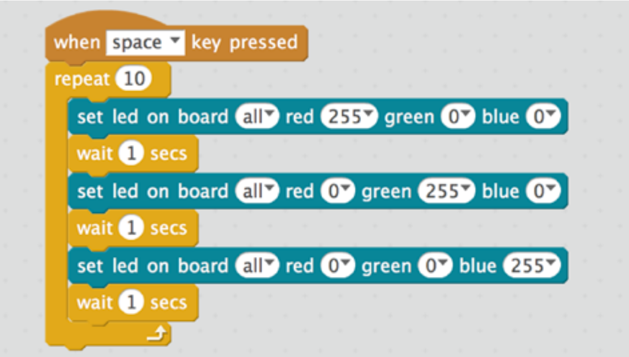
```
when space key pressed
set led on board all red 255 green 0 blue 0
wait 1 secs
set led on board all red 0 green 255 blue 0
wait 1 secs
set led on board all red 0 green 0 blue 255
wait 1 secs
```

Figure 20. Image of 3 light sequences

Ask children to test their predictions by making the code in mBlock and testing. ***Where they correct?***

While the groups are testing, extend any quick finishers by asking them to add a block that will allow the sequence to keep happening.

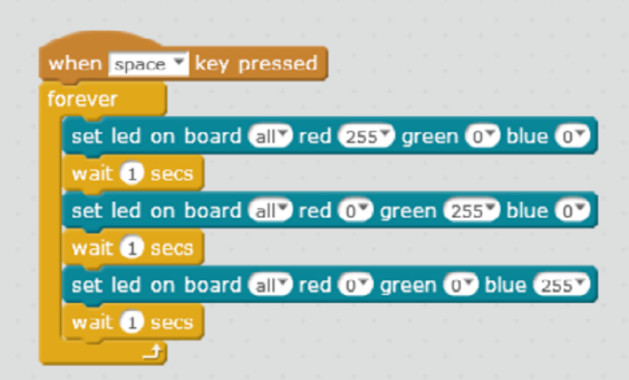
Once all the children have successfully made a simple 3 light sequence, ask then **what would I need to add to the code to make it repeat 10 times?** Children to discuss. Demonstrate adding in a report loop from the Control blocks. So the code looks like the example below.



```
when space key pressed
repeat 10
    set led on board all red 255 green 0 blue 0
    wait 1 secs
    set led on board all red 0 green 255 blue 0
    wait 1 secs
    set led on board all red 0 green 0 blue 255
    wait 1 secs
```

Figure 21. Image of 3 light sequence in a repeat loop

Ask children how can I keep the sequence going forever? Children to discuss and give ideas. Demonstrate swapping out the repeat 10 times block for a forever block from the Control blocks, as shown in the example below.



```
when space key pressed
forever
    set led on board all red 255 green 0 blue 0
    wait 1 secs
    set led on board all red 0 green 255 blue 0
    wait 1 secs
    set led on board all red 0 green 0 blue 255
    wait 1 secs
```

Figure 22. Image of 3 light sequence in a forever loop

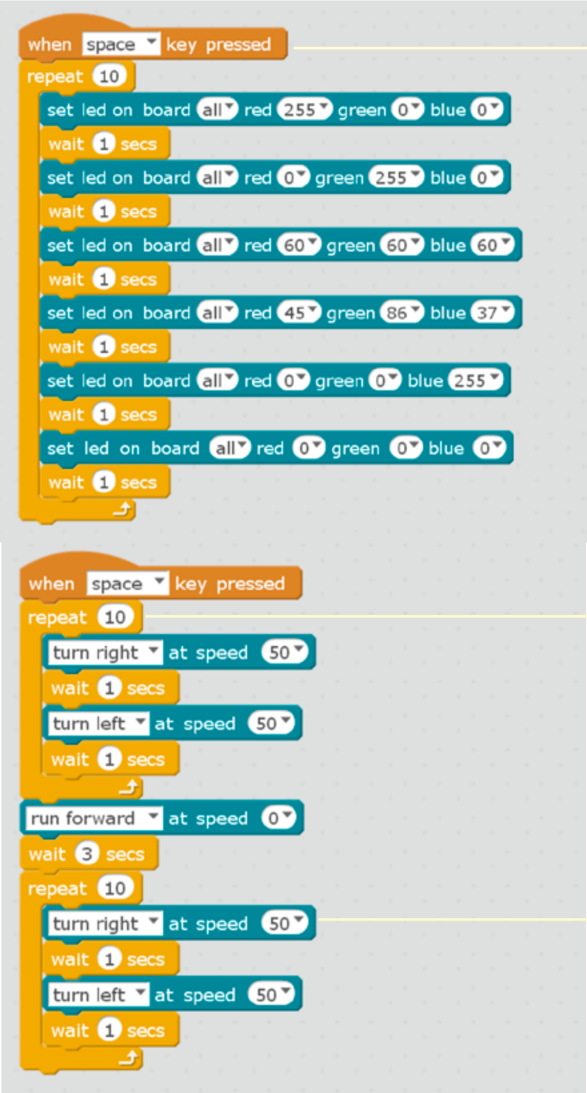
Now the children know how to flash the lights on the mBot.

It is time to set the groups/pairs a challenge, we are going to program the mbot to dance along to a piece of music., flashing it’s disco lights.

The mBot will need to move on the spot and flash its LEDs in a light sequence that has at least 6 different colours and repeats 10 times.

Choose a simple piece of music that the children will use. This site has some nice royalty free tracks https://www.purple-planet.com/ though you can use anything you wish.

An example of what the children could produce is below.



```
when space key pressed
repeat 10
    set led on board all red 255 green 0 blue 0
    wait 1 secs
    set led on board all red 0 green 255 blue 0
    wait 1 secs
    set led on board all red 60 green 60 blue 60
    wait 1 secs
    set led on board all red 45 green 86 blue 37
    wait 1 secs
    set led on board all red 0 green 0 blue 255
    wait 1 secs
    set led on board all red 0 green 0 blue 0
    wait 1 secs

when space key pressed
repeat 10
    turn right at speed 50
    wait 1 secs
    turn left at speed 50
    wait 1 secs
run forward at speed 0
wait 3 secs
repeat 10
    turn right at speed 50
    wait 1 secs
    turn left at speed 50
    wait 1 secs
```

This will repeat 10 times a sequence of LEDs flashing. There will be 6 colours and a each will be shown for 1 second.

This will make the mBot move left and right 10 times.

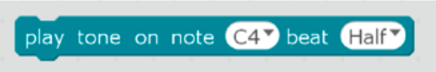
This will make the mBot spin to the right.

Figure 23. Image of possible answer

Differentiation

- Less able children could be given a slightly different challenge of just making the LEDs flash in a sequence. You could also reduce the number of colours from 6 to 4.

- More able children would be expected to create a more complex program. You could extend the challenge by asking them to have the light sequence have at least 10 colours. Also they could be encouraged to explore the play tone block.

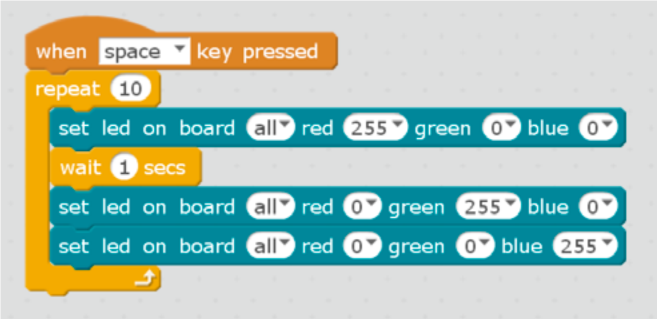


```
play tone on note C4▾ beat Half▾
```

This is in the Robots blocks and will allow the mBot to sing. In practice it can be a bit quiet in a noisy classroom.

Plenary - 10 minutes.

Show the piece of code below. Ask the class, **What is wrong with this code? Can they predict what it will do?** *Answer: The LEDs will only go red then blue, the LEDs won't go green as it happens too fast for us to see, because there is no wait after the change to green.*



```
when space ▾ key pressed
repeat 10
    set led on board all▾ red 255▾ green 0▾ blue 0▾
    wait 1 secs
    set led on board all▾ red 0▾ green 255▾ blue 0▾
    set led on board all▾ red 0▾ green 0▾ blue 255▾
```

Figure 24. Image of incorrect code

Ask the class, **What where the inputs and outputs that we used in the task?** *Answer: Inputs – Keyboard to program and start the dancing, Outputs – LEDs, Motors, Buzzer.*

Ask groups to share the mBots dance moves. You can technically get all the mBots to ‘dance’ at the same time, but it is probably more practical to choose a few groups at a time. Which dance is most convincing? If any group makes the mBot move in an interesting way get them to share their code with the class and explain it.

8

LESSON 3
MAKING ME SMILE - FURTHER UNDERSTANDING OF INPUT AND OUTPUTS

Overview

In this lesson we will be learning how to add and use the LED matrix with the mBot. We will create an animated expression for the mBot.

NC Links

- 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

Lesson Objective

I can program the mBot to display text or an image on the LED display.

Intro - 20 minutes

Ask the children to explain, What are the input and outputs on the mBot?

Input	Output
Children may not get all of these as some can't be seen through the mCore cover	
Follower Sensor	Motor
Ultrasonic Sensor	**On the board:** LEDs Buzzer IR Transmitter
On the board: Button IR Receiver Light Sensor	

Can they explain what they do?

Follower Sensor	A sensor that allows the mBot to recognise light and dark, so can follow a line.
Ultrasonic Sensor	A sensor that allows the mBot to sense if something is in front of it and how far away it is.
Motor	Spins.
On the board: Button IR Receiver Light Sensor LEDs Buzzer IR Transmitter	Button - Push to turn on. LEDs - Lights that can be set to any colour using RGB values. (See page 40) Buzzer - Makes a sound. IR Transmitter - Sends a message via Infrared. IR Receiver - Detects when an Infrared message as been sent.

Introduce the LED display to the class.

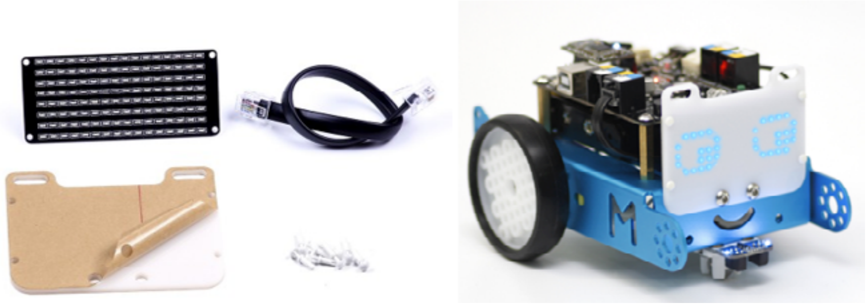


Figure 25. Image of LED Matrix as parts and attached.

Explain that the LED display is an output device that we are going to add to the mBot. (You may have decided to preinstall these before the lesson.) In order to use the LED Matrix you need to swap out the Ultrasonic sensor.

Start by removing the two screws at the bottom of the Ultrasonic Sensor. Unplug the cable from the rear of the Ultrasonic Sensor. Put the Ultrasonic Sensor to one side. Take the LED Matrix and replug the cable into the back of it. Then attach the LED Matrix using the same screws that you just removed. The piece of plastic that is in front of the actual display is know as a diffuser. The LEDs are very bright so it will cut down the brightness but also allows images and text to display more clearly.

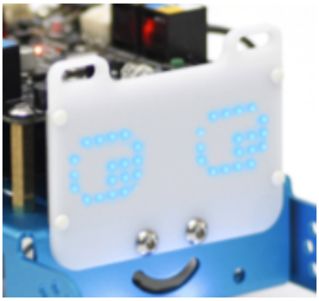


Figure 26. Image of Close Up of LED Matrix attached to mBot.

Figure 27. Image of Close Up of Ultrasonic Sensor attached to mBot.

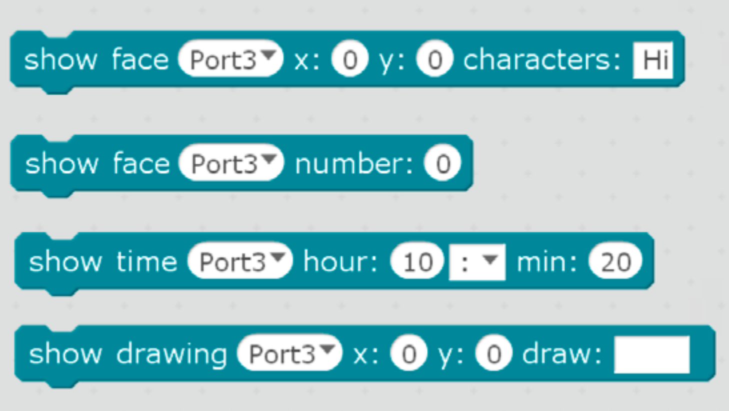
Explain that we are going create a program that will make the LED Matrix display something.

Main - 25 minutes

Set class a challenge, can they make LED Matrix display some text, number or perhaps a picture when they push a key on the keyboard?

Where will you be looking for the blocks? What clues will you be looking for in the blocks? Get children to discuss and decide on some keywords they will be looking for within the Robots section.

These are the only 4 blocks that relate to the LED Matrix.



```
show face Port3▾ x: 0 y: 0 characters: Hi
show face Port3▾ number: 0
show time Port3▾ hour: 10 : ▾ min: 20
show drawing Port3▾ x: 0 y: 0 draw:
```

Figure 28. Image of LED Matrix blocks.

The 1st one will display a short 3 characters piece of text. You can type longer words but it won't display it all. It does not scroll.

The 2nd one will display any number up to 9999, it can't display any larger numbers, if you try it will just display 9999.

The 3rd one will display a time, hours and minutes.

The 4th one is the most useful and will allow you to draw a picture which can be displayed. Click on the white rectangle and this box will appear. See Figure 29 on the next page.

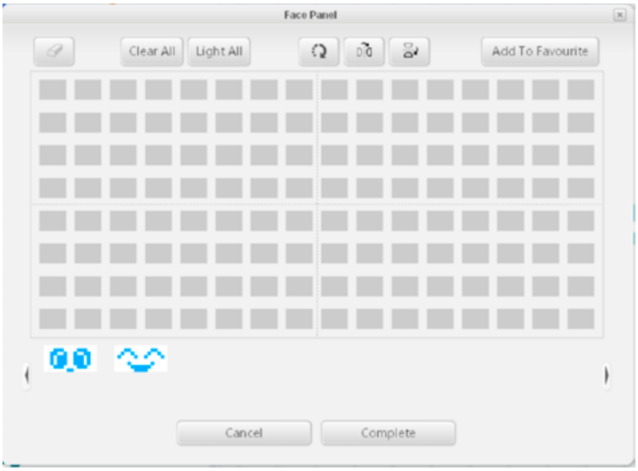
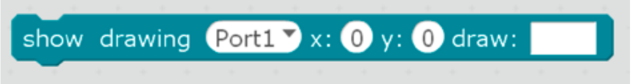


Figure 29. Image of LED Matrix Options from drawing block.

Click on the lights that you want to turn on. You have some preset faces you can use and you can Add to Favorite images you create. Click Complete to store the image on the block. The white rectangle will have a tiny preview of what should appear on the LED Matrix.

Give each group/pair an mBot and paired Wireless Dongle. Children will need to connect mBot to mBlock software before they can can start the challenge. See page 19 for instructions on how to connect the mBot to the mBlock software.



```
show drawing Port1 x: 0 y: 0 draw:
```

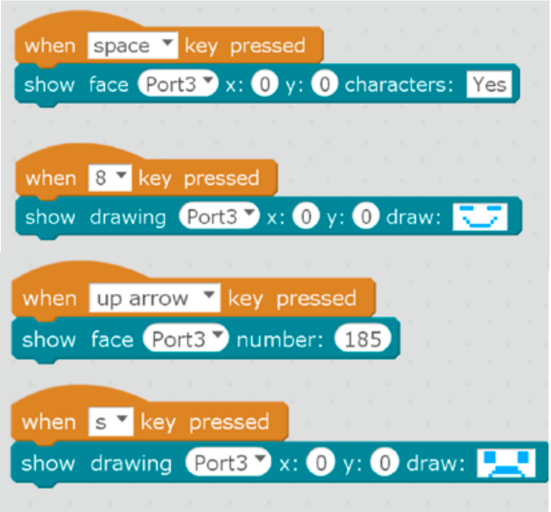
Figure 30. Image of drawing block for LED Matrix.

Remind children of the challenge. **Can they make LED Matrix display some text, number or perhaps a picture when they push a key on the keyboard?**

As children work encourage them to try all the blocks but everyone should spend the most time exploring the image creation block.

During the lessons ask children to share success with each other. **Can you tell everyone how you did a particular task?**

Some possible solutions:

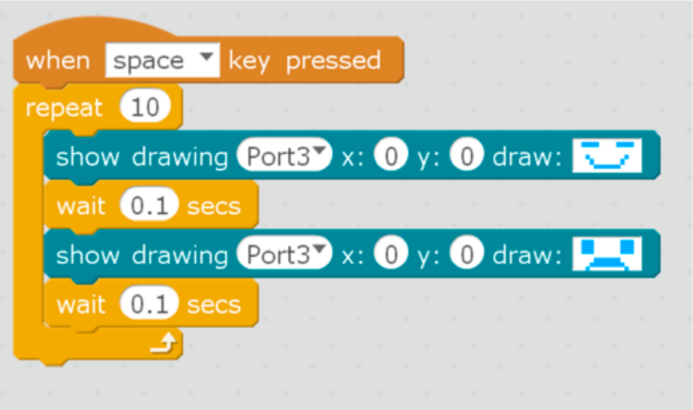


```
when space key pressed
show face Port3 x: 0 y: 0 characters: Yes

when 8 key pressed
show drawing Port3 x: 0 y: 0 draw:

when up arrow key pressed
show face Port3 number: 185

when s key pressed
show drawing Port3 x: 0 y: 0 draw:
```

Figure 31. Image of possible solutions for using LED Matrix Blocks.

After 10 minutes or so move the children on by posing a new challenge. **Can you create a simple animation?** Children to discuss what an animation is. A series of images that are switched between really quickly. How could we program that? I have two images of expressions what do I need to add?

Possible solution:



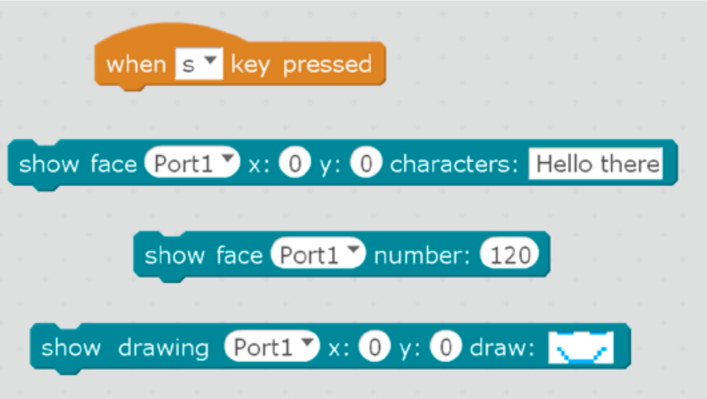
```
when space key pressed
repeat 10
    show drawing Port3 x: 0 y: 0 draw:
    wait 0.1 secs
    show drawing Port3 x: 0 y: 0 draw:
    wait 0.1 secs
```

Figure 32. Image of possible solution for creating animation using LED Matrix.

Children to continue to explore blocks and try and make a simple animation.

Differentiation

- Less able children could be given a saved file with all the necessary blocks already dragged out. All they need to do it click them together and then spend more time exploring how changing the values affects what is displayed on the LED Matrix. For example:



```
when s key pressed

show face Port1 x: 0 y: 0 characters: Hello there

show face Port1 number: 120

show drawing Port1 x: 0 y: 0 draw:
```

Figure 33. Image of example blocks.

- More able children could be encouraged to add expressions to their remote control program from Lesson 1. Can they get mBot look left and right as it turns? Can mBot look scared going backwards can mBot look excited moving forwards?

Plenary - 5 minutes

How can you create a simple animation? Get some pairs/groups to show what they have animated. **Can you explain your code?** Get them to explain their code.

Top Tips

If the LED matrix doesn’t work, check that you have it plugged into Port 3. This is the same Port that the Ultrasonic sensor was plugged into (Port 3). Also remember to check the batteries the LED Matrix is very battery hungry.

The display also remembers the last instruction sent to it. So at the end of lessons children should clear the display.

space key pressed
on board all red 255 green 0 blue 0
secs
on board all red 0 green 255 blue 0
secs
on board all red 0 green 0 blue 255
secs
when w key pressed
run forward at speed 100
when w key released
run forward at speed 0
when a key pressed
turn left at speed 100
when a key released
turn left at speed 0
when space key pressed
run forward at speed 0
mBot
educational robot kit
Makeblock
Construct Your Dreams
when clicked
set speed to 250
play sound eat until done
point in direction 90
set pen color to
pen down
go to x: 22 y: -17
forever
say light intensity
run forward at speed
Great tool for beginners to learn
Graphical Programming, Electronics and Robotics
www.makeblock.cc

LESSON 4
SENSING THE WORLD - AN INTRODUCTION TO USING AN ULTRASONIC SENSOR

Overview

In this lesson we will be learning how to use the ultrasonic sensor on the mBot. We will get the mBot to stop moving when it meets and object.

NC Links

- 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

Lesson Objective

I can program the mBot to stop when it reaches an obstacle.

Intro - 20 minutes

Ask the children to explain, **What are the input sensors on the mBot?**

Follower Sensor	A sensor that allows the mBot to recognise light and dark, so it can follow a line.
Ultrasonic Sensor	A sensor that allows the mBot to sense if something is in front of it and how far away it is.

Can they explain what they do?

Explain that we are going create a program that will stop the mBot when it reaches a wall or obstacle.

Ask the class, What sensor do we need to use to make an mBot stop moving when it reaches a wall/obstacle? Children to discuss and hopefully suggest the Ultrasonic sensor.

Discuss as a whole class, **what are the instructions we need to give to the mBot to stop when it reaches an obstacle?** Get children discuss in small groups, remind them to focus on what needs to happen, and not get stuck with the correct language. Record the classes ideas on a whiteboard. These ideas may be a set of instructions or an explanation of what the mBot will do. Below is how you could scaffold writing the instructions though questioning.

Ask - So what does the mBot need to do?

If the mBot gets close to a wall it stops.

Ask - What was it doing before that?

The mBot moves forward.
If the mbot gets close to a wall it stops.

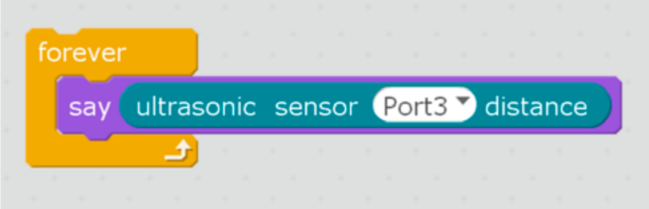
Ask - What is causing it to stop?

The mBot moves forwards.
If the mBot gets close to a wall, the ultrasonic sensor will detect it.
The ultrasonic sensor tells the mBot to stop.

Through this process the children have written an algorithm, a written solution to a problem.

Explain that now we need the algorithm we have to turn it into code.

Show this code to the children. Want do you think it will do? Give children time to discuss. *Answer: The panda sprite will say the distance from an object to the ultrasonic sensor.*



```
forever
  say ultrasonic sensor Port3 distance
```

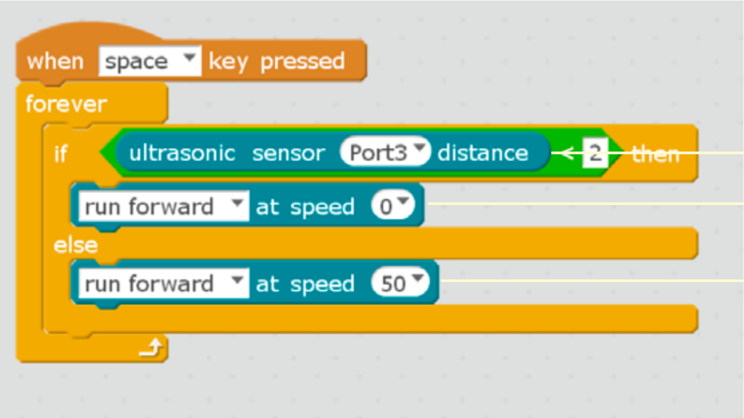
Image 34. Image of distance sensor code running

While the children are discussing, you will need to connect the mBot to the mBlock software before you can demonstrate the code. See page 19 for instructions on how to connect the mBot to the mBlock software. Click on the blocks to run the code, demonstrate the code working by moving your hand towards and away from the ultrasonic sensor, the value will get smaller the closer you hand is to the sensor.

Top Tip
If the sensor doesn't seem to work, check the following. What port is the sensor on the robot plugged into? Normally Ultrasonic sensor is plugged into Port 3 and the Line Follower sensor is plugged into Port 2. Check on the block what Port has been selected. They MUST match.

As a class create the following code linking back to the algorithm that you created as a class.

The mBot moves forwards.
If the mBot gets close to a wall, the ultrasonic sensor will detect it.
The ultrasonic sensor tells the mBot to stop.



```
when space key pressed
forever
    if ultrasonic sensor Port3 distance < 2 then
        run forward at speed 0
    else
        run forward at speed 50
```

If the mBot gets close to a wall, the ultrasonic sensor will detect it.

The ultrasonic sensor tells the mBot to stop.

The mBot move to forward.

Figure 35. Image for example code and explanation

What Is the difference between an algorithm and coding?

An algorithm is the plain language solution to a problem. It is how would explain it in plain spoken English to another person.

Coding is how you get a computer to understand the solution. You have to change the solution so that it is written in a language that the computer understands, in this case Scratch. So the coding will differ language to language but the algorithm will be the same.

Main - 25 minutes

Give out the mBots and the Wireless Dongles to pairs/groups.

Get children to copy the example code and programme their mBot to move and stop as close as possible to a wall/obstacle. Get the groups/pairs to consider some of the following questions. **How close can you get the mBot to stop? What is the fastest it can move and still detect an obstacle?**

Differentiation

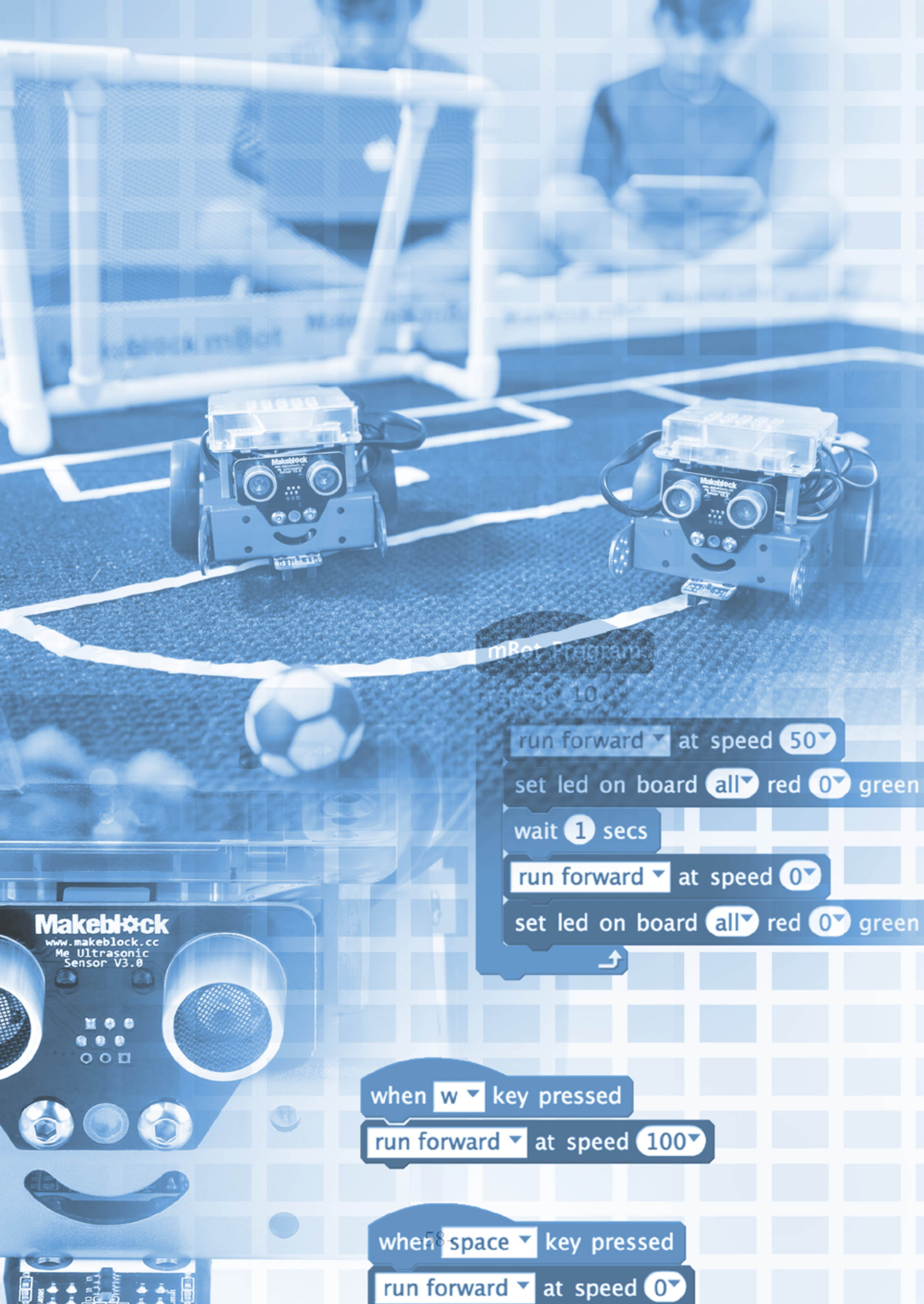
- Less able children could be given a file with all the code already completed. They can then spend more time exploring how changing the values to get the mBot stopping as close a possible to the obstacle.
- More able children could be encouraged to explore what could the mBot do instead of just stopping when it is close to an obstacle. **Could the mBot be programmed to avoid the obstacle?**

Plenary - 5 minutes

Ask groups to share what values were most successful? What did you have to realise about the speed of the mBot? Discuss. *Answer: Values must be between 1-2. Any lower and the sensor isn't accurate enough. Speed has to be slow to allow the mBot time to realise it is close to an obstacle. Too fast and it will hit the obstacle. The slowness is due to the time it takes for the mBot to sense, send the value to the computer, be processed by the program, and then send the right instruction back to the mBot.*

Top Tips

- Obstacles need to be of a certain height to be detected. I have used books and dictionaries in the past to make obstacles. Test the objects you are going to use before the lesson. A nice idea is to create a wall round the edge of a tabletop using books.
- The speed of the mBot needs to be kept quite slow as you need to give the mBot time to process the sensor information and act on it. If it moves too fast the mBot will stop after it hits the obstacle.


mBot Program
repeat 10
run forward at speed 50
set led on board all red 0 green
wait 1 secs
run forward at speed 0
set led on board all red 0 green
when w key pressed
run forward at speed 100
when space key pressed
run forward at speed 0
Makeblock
www.makeblock.cc
Me Ultrasonic
Sensor V3.0

10

LESSON 5 MOVING ALL BY YOURSELF - AUTONOMUS MOVEMENT USING AN ULTRASONIC SENSOR

Overview

In this lesson we will be developing our skills of using the ultrasonic sensor. We will get the mBot to move around by itself avoiding obstacles.

NC Links

- 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

Lesson Objective

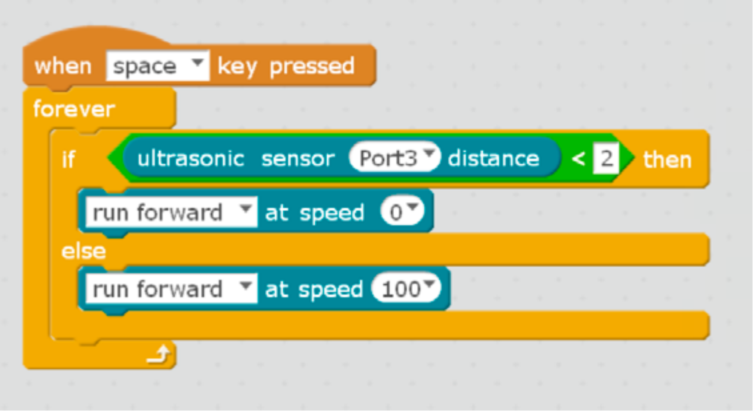
I can program the mBot to move around avoiding obstacles.

Intro - 20 minutes

Revisit last lessons work. Display the code below. **Can you explain the following code?**

What was the task/problem that this code was designed to solve?

Answer – to stop as close as possible to an obstacle.



```
when space key pressed
forever
    if ultrasonic sensor Port3 distance < 2 then
        run forward at speed 0
    else
        run forward at speed 100
```

Figure 36. Image of example code

Can anyone remember the algorithm that was used to create this code? Children to discuss and if possible get them to share what they remember of last lessons algorithm.

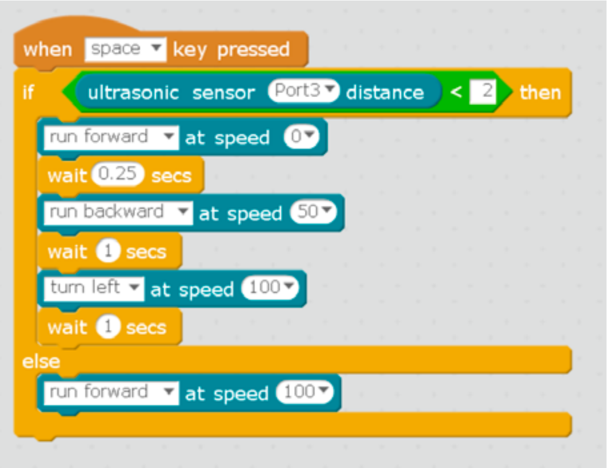
Display last lessons algorithm on a screen or interactive whiteboard.

The mBot was moving forwards.
If the mBot gets close to a wall, the ultrasonic sensor will detect it.
The ultrasonic sensor tells the mBot to stop.

This lesson’s problem is **how to change this code to allow the mBot to move around by itself avoiding obstacles.**

Before the lesson have set up on a table/floor an area with obstacles around the edge.

Have a mBot connected to the mBlock software with the following program ready to run.



```
when space key pressed
if ultrasonic sensor Port3 distance < 2 then
    run forward at speed 0
    wait 0.25 secs
    run backward at speed 50
    wait 1 secs
    turn left at speed 100
    wait 1 secs
else
    run forward at speed 100
```

Figure 37. Image of movement code

Activate the program by hitting the spacebar and demonstrate the mBot moving around the area without hitting any of the walls.

Get children to discuss what they are seeing. **What is the algorithm that is being followed?** It is similar to last weeks but with one important addition. **What do we need to add to our original algorithm to make the mBot do more than stop?** *Answer: After stopping it reverses and then turns.*

Discuss as a group and decide as a class what the new algorithm is. Add highlighted section to last lessons algorithm. Display on a screen or interactive whiteboard so class can refer to it.

The mBot is moving forwards.
If the mBot gets close to a wall, the ultrasonic sensor will detect it.
The ultrasonic sensor tells the mBot to stop **and then reverse and turn.**

Main - 30 minutes

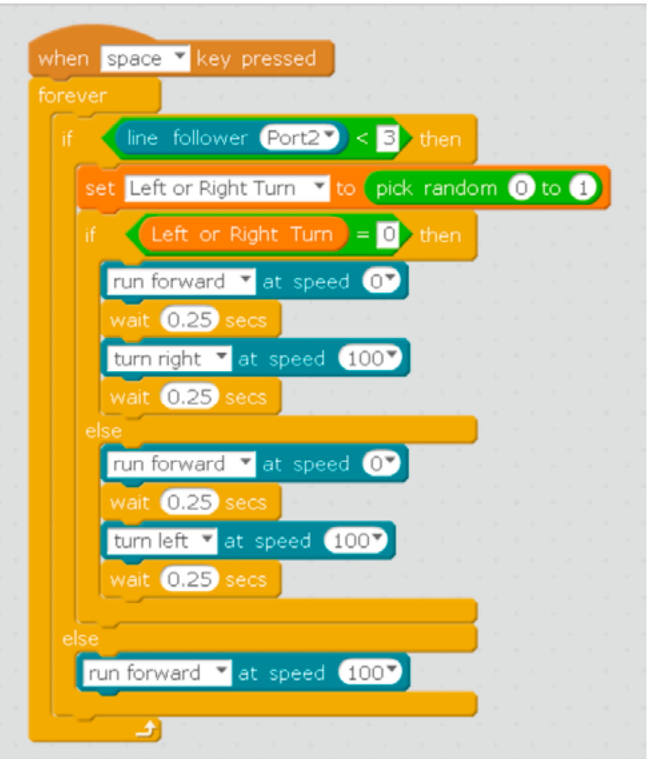
Children to work in groups/pairs and complete the task, to make the mBot to move around by itself avoiding obstacles. Children to use their code from last week as a starting point and the algorithm we have just discussed and create an solution to the problem.

The solution will be something like the example you used to demonstrate how the mBot should behave on the previous page.

Please note: Each robot might turn different amounts, see the Top Tip at the end of this chapter for more information.

Differentiation

- Less able children could be given a file with all the code already completed. They can then explore changing the values of motor speed and wait times to allow the mBot to turn correctly.
- More able children were asked to look at avoiding obstacles last lesson. In order to further extend their learning ask, Should the mBot always turn left? Can they randomise if the mBot turns left or right once it has stopped? Below is a possible solution.



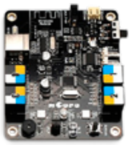
```
when space key pressed
forever
  if line follower Port2 < 3 then
    set Left or Right Turn to pick random 0 to 1
    if Left or Right Turn = 0 then
      run forward at speed 0
      wait 0.25 secs
      turn right at speed 100
      wait 0.25 secs
    else
      run forward at speed 0
      wait 0.25 secs
      turn left at speed 100
      wait 0.25 secs
  else
    run forward at speed 100
```

Figure 38. *This code randomly picks a number when the ultrasonic sensor gets too close to an obstacle. Depending on the number either the code that turns right is selected or the code that turns left.*

Plenary - 10 minutes

Ask the class, What can we make the mBot do? *Answer: make it move, use the ultrasonic sensor, make it flash LEDs.*

What are the inputs and outputs on an mBot?

Input		Output	
Children may not get all of these as some can't be seen through the mCore cover			
	Follower Sensor		Motor
	Ultrasonic sensor		On the board: LEDs Buzzer IR Transmitter
	On the board: Button IR Receiver Light Sensor		

What sensor haven't we looked at yet? *Answer: The Line follower*

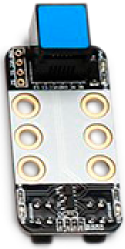


Figure 38- Image of line follower sensor.

What do you think you can do with this sensor? *Answer: detect a dark line, detect an edge, it works by shining IR light and seeing what reflects, dark colours absorb the IR light.*

What do you think you can do with this sensor? *Answer: detect a dark line, detect an edge, it works by shining IR light and seeing what reflects, dark colours absorb the IR light.*

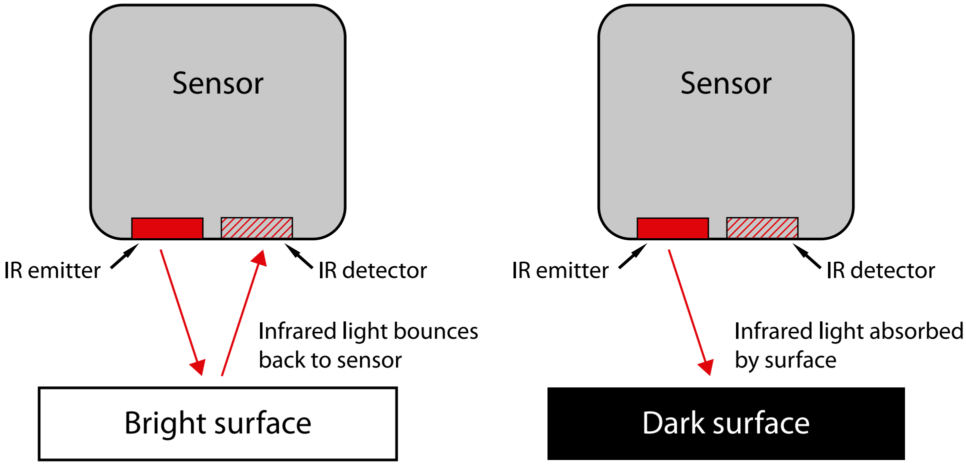


Figure 39. *Optical sensor diagram.*

Top Tip

- You may find that each robot will turn different amounts. You cannot tell the motors to move a set number of degrees as the motors are very simple. You can just tell them to turn on and off for a set amount of time. One solution may not work for all the mBots as the internal parts of the motor will differ as well as the amount of charge in the batteries. To adjust the turn you will need to carefully adjust the wait timer, generally you are looking for a value less than one second. Remember, the wait command, allows the mBot to run the previous command for a set time and then move onto the next instruction.

11

LESSON 6
DON'T FALL - UNDERSTANDING HOW TO USE A LINE FOLLOWER SENSOR

Overview

In this lesson we will be learning how to use the line following sensor. We will apply knowledge we have gained in previous lessons to make the mBot move around a table avoiding the edge.

NC Links

- 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

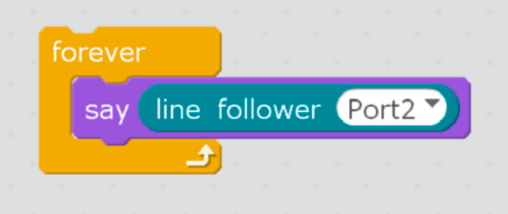
Lesson Objective

I can program the mBot to detect the edge of a table.

Intro - 20 minutes

Explain that last lesson we discussed that an Line follower sensor can detect an edge or a line.

Show the class this code.



```
forever
  say line follower Port2
```

Figure 40- Image of code

What do you think it will do? Relate it back to the example with the ultrasonic sensor two lessons ago if needed. Children to discuss. *Answer - the panda will display a value, according to what the line follower detects. It will be a number between 0, 1, 2 or 3.*

Have a mBot connected to the mBlock software with the program ready to run and demonstrate how this works. This can be done with a table edge or a thick black line. With each mBot there is a poster that has a line on it which can be used for line following activities.

Focus for now on making it detect an edge by both lights going out. Do not use your hand as covering the whole sensor prevents them from detecting anything, it will show nothing being detected.

You will notice that normally on the back of the line follower two red LEDS are on.

What the numbers and lights mean:
0 = Both lights off. – Detects a line/edge.
1 = Left light off - Detects a line/edge to the left.
2 = Right light off - Detect a line/edge to the right.
3 = Both lights on. – Nothing detected.

Explain that today are going to make the mBot avoid the table edge. The class will be using what they know about the ultrasonic sensor and making the mBot avoid obstacles.

Why is the ultrasonic sensor not useful when detecting an edge? *Answer – ultrasonic sensors need something to bounce sound off, an edge has no surface.* **Why do we need to use other sensors?** *Answer - Different sensors do different jobs. Human use a range of senses to interact with the world, so do robots.*

This lesson's problem is **how to program the mBot to move around the table, but avoiding falling off the edge.**

Ask the class, **What is the algorithm that the mBot needs to follow? What does the mBot need to do?** Children can discuss in their groups and record the algorithm for using a line follower sensor. You could display the algorithm for the ultrasonic sensor as a starting point if they get stuck.

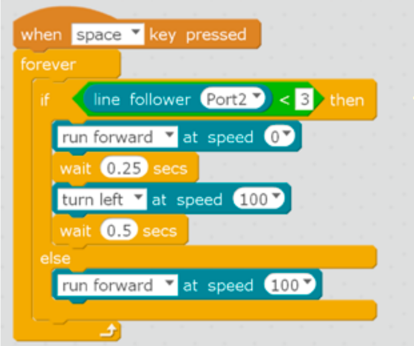
> *The mBot is moving forwards.*
> *If the mBot gets close to a wall, the ultrasonic sensor will detect it.*
> *The ultrasonic sensor tells the mBot to stop and then reverse and turn.*

Get the groups to share their ideas and as a class agree on a shared algorithm for the whole class to refer to, it should be something like this:

> *The mBot is moving forwards.*
> *If the mBot gets to an edge the line follower sensor will detect it.*
> *The line follower sensor tells the mBot to stop and then backup and turn away from the edge.*

You already have an mBot connected to the mBlock software, while the children were discussing the algorithm, write the program below to demonstrate an example of an mBot running code that makes it avoid the edge of a table. The mBot will need to placed on a table.

Ask the children, **Does the behavior you have seen match be your algorithm? Do you need to make any changes?** Speed is one thing that children may notice here, just like with the ultrasonic sensor, you have to be going slow enough for the line-follower sensor to detect the edge and the code to react.



```
when space key pressed
forever
  if line follower Port2 < 3 then
    run forward at speed 0
    wait 0.25 secs
    turn left at speed 100
    wait 0.5 secs
  else
    run forward at speed 100
```

If the line follower detecs anithing stop and then turn. **Remember 3 is the only value that shows that nothing is detected.**

Figure 41- Example code to avoid an edge

Main - 30 minutes.

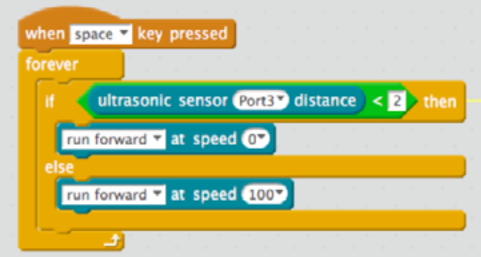
Set children off on the problem of program the mBot to move around the table, but avoiding falling off the edge. Children should look at their code from the ultrasonic sensor as a starting point. The solution will be something like the example above.

Top Tip

- You may find that each robot will turn different amounts. You cannot tell the motors to move a set number of degrees as the motors are very simple. You can just tell them to turn on and off for a set amount of time. One solution may not work for all the mBots as the internal parts of the motor will differ as well as the amount of charge in the batteries. To adjust the turn you will need to carefully adjust the wait timer, generally you are looking for a value less than one second. Remember, the wait command, allows the mBot to run the previous command for a set time and then move onto the next instruction.

Differentiation

- Less able children could be given a file with the the simple ultrasonic sensor from last lesson already completed. They can work how to adjust the code to make it use the Line-follower sensor and make it stop at the edge of a table.



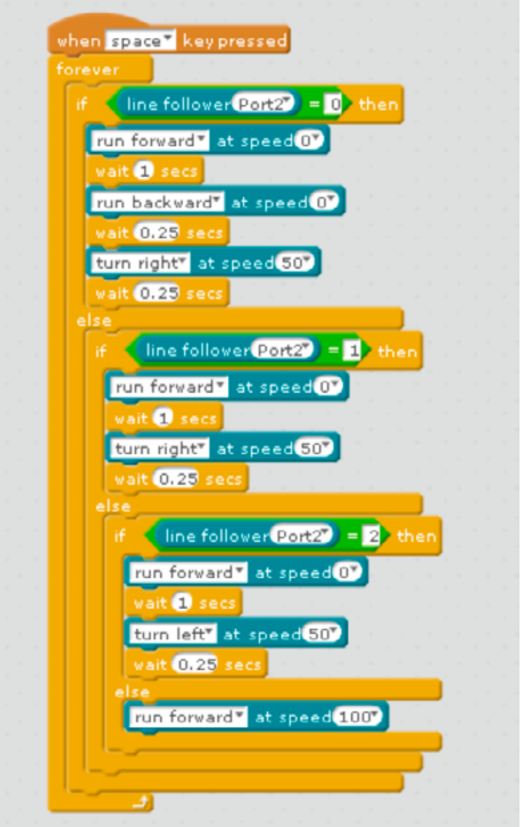
```
when space key pressed
forever
  if ultrasonic sensor Port3 distance < 2 then
    run forward at speed 0
  else
    run forward at speed 100
```

Figure 42. Image of incorrect code. Answer: You just need to swap the ultrasonic sensor block for the line follower block and adjust the value to 3.

- Higher ability children could be encouraged to explore how they can they use all the values from the line follower sensor to make the robot turn more accurately? In the example if anything is detected on the line flower, the mBot should stop and turn. This could make the mBot turn off the table in some cases. If the sensor shows that the left side if off the table, it should back up to the right to get away from the edge.

- Remember what the numbers mean:
  - 0 = Both lights off. – Detects a line/edge.
  - 1 = Left light off - Detects a line/edge to the left.
  - 2 = Right light off - Detect a line/edge to the right.
  - 3 = Both lights on. – Nothing detected.

A possible solution:



```
when space key pressed
forever
  if line follower Port2 = 0 then
    run forward at speed 0
    wait 1 secs
    run backward at speed 0
    wait 0.25 secs
    turn right at speed 50
    wait 0.25 secs
  else
    if line follower Port2 = 1 then
      run forward at speed 0
      wait 1 secs
      turn right at speed 50
      wait 0.25 secs
    else
      if line follower Port2 = 2 then
        run forward at speed 0
        wait 1 secs
        turn left at speed 50
        wait 0.25 secs
      else
        run forward at speed 100
```

Figure 43. Line following solution example

Plenary - 10 minutes

Ask the children, **What have we over the last few weeks what we have programed the mBot to do?** Give children some time to discuss. *Answer: Made it avoid edges and objects, controlled it with a keyboard, etc…*

If we combine the programs to avoid an edge and the program to avoid obtacles. **What will mBot be able to do? Where will it be able to travel? Give children time to discuss. Could it roam around the school? What else does it need to able to do for it to move anywhere?** *Answer: possible answers, more control over its motors, a high a low setting, ability to get down stairs, a wider ultrasonic view, more sensors, etc…*

Show image of The Curiosity rover. **Does anyone know what this is?** *Answer: a robot on mars.*

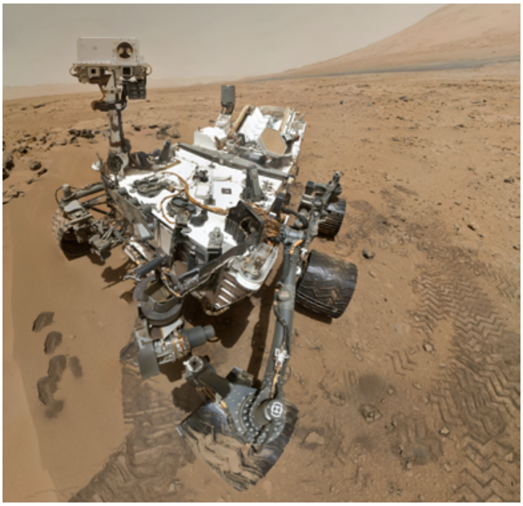


Figure 44. Image of curiosity rover.

Explain that this real robot needs to be able to 'think' for themselves. It has to be able to transverse a wide range of terrain by itself, it can't ask for help from Earth as it takes far too long for instructions to be sent and received. If you have time this video showcasing how teh robot moved across Mars during its first 5 years is worth a look. http://youtu.be/O0nPFaBU98k

12

LESSON 7
REMOTE CONTROL - HOW TO REPROGRAM THE CONTROLLER

Overview

In this lesson we will learn how to program the mBot to respond to the IR remote control.

NC Links

- 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

Lesson Objective

I can program the mBot to respond to the IR controller?

Intro - 10 minutes

Show children the IR controller. **What do you think this is for?** Children to discuss.

Show the image below and explain that this shows what the controller did when the mBot was first built. It doesn't work anymore and we are going to work out how to **make the IR remote control to control the mBot again.**

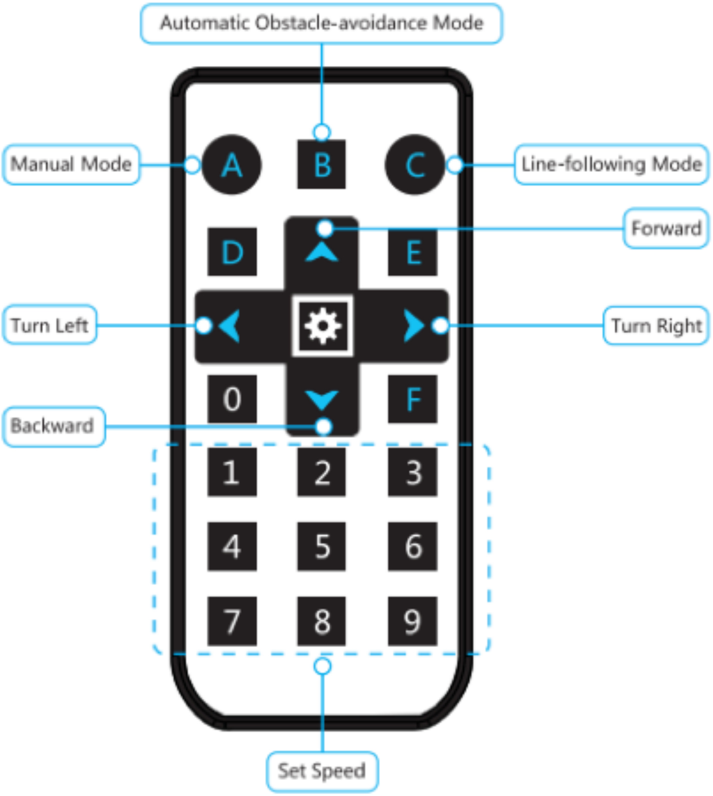


Figure 45. Image of IR Remote Control.

Give each pair/group a controller. Get them to discuss and record the algorithm for making the controller work. **What do they want to happen when they push a button?**

Some groups will come up with:

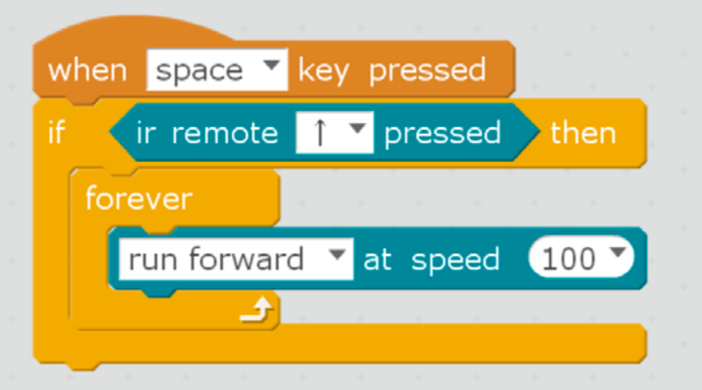
If up is pressed the robot moves forward
If down is pressed the robot moves backward
If left is pressed the robot turns left
If right is pressed the robot turns right

Some things they may not be sure about, Set Speed for example. This will refer to the speed setting of the mBot. For now ignore the A, B and C button.

Get the children to share some ideas and make sure that all the groups have an idea of how to start.

Ask the class, **Is the controller an input or an output?** *Answer: an input*

Show the following code. **What does this do?** Children to discuss and give suggestions.



```
when space key pressed
if ir remote ↑ pressed then
  forever
    run forward at speed 100
```

Figure 46. Example code for moving forward using remote.

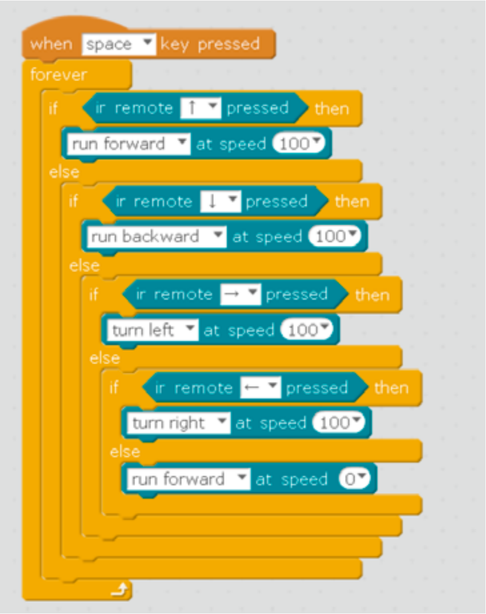
Answer: When you push the Up arrow on the controller it will start to move forward. It won't stop.

Main - 40 minutes

Give out mBots and Wireless Dongles to each group/pair. Get them to connect the mBot to the mBlock software so they are ready to start.

Set children off on the problem to **make the IR remote control to control the mBot.**

This is a possible solution.

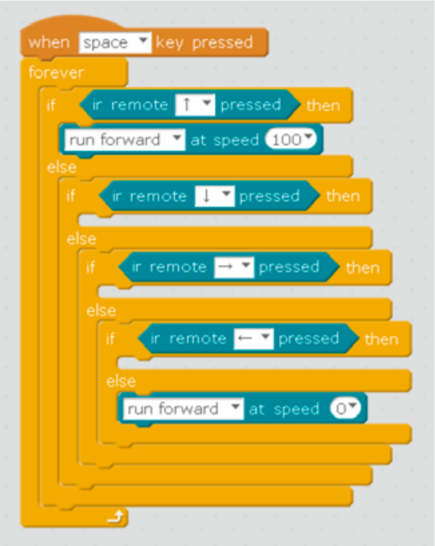


```
when space key pressed
forever
  if ir remote ↑ pressed then
    run forward at speed 100
  else
    if ir remote ↓ pressed then
      run backward at speed 100
    else
      if ir remote → pressed then
        turn left at speed 100
      else
        if ir remote ← pressed then
          turn right at speed 100
        else
          run forward at speed 0
```

Figure 47. Example code for remote control.

Differentiation

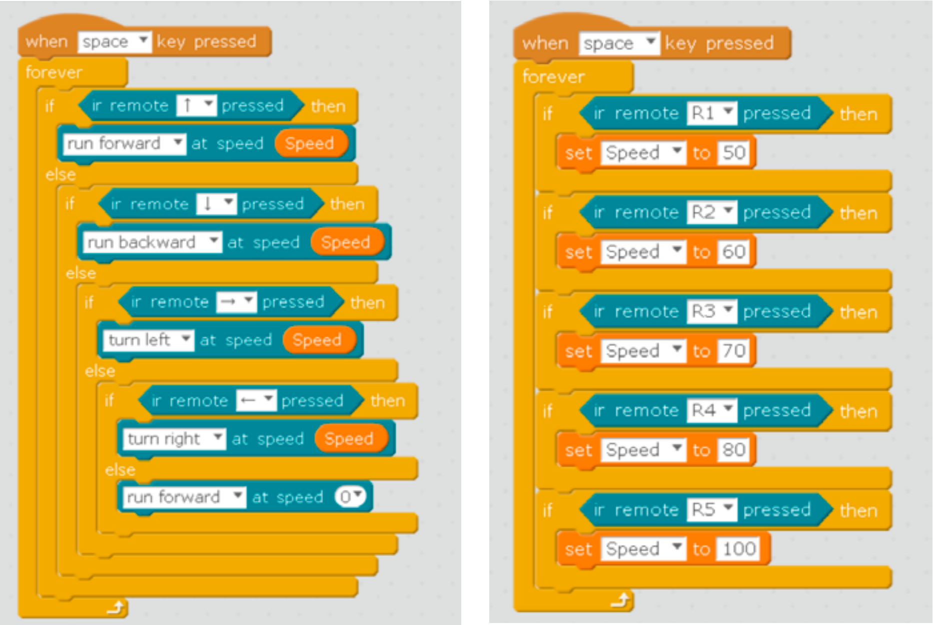
Less able children could be given a partly completed saved program. They will have to complete it. For example:



```
when space key pressed
forever
  if ir remote ↑ pressed then
    run forward at speed 100
  else
    if ir remote ↓ pressed then
    else
      if ir remote → pressed then
      else
        if ir remote ← pressed then
        else
          run forward at speed 0
```

Figure 48. Partly completed solution for remote control.

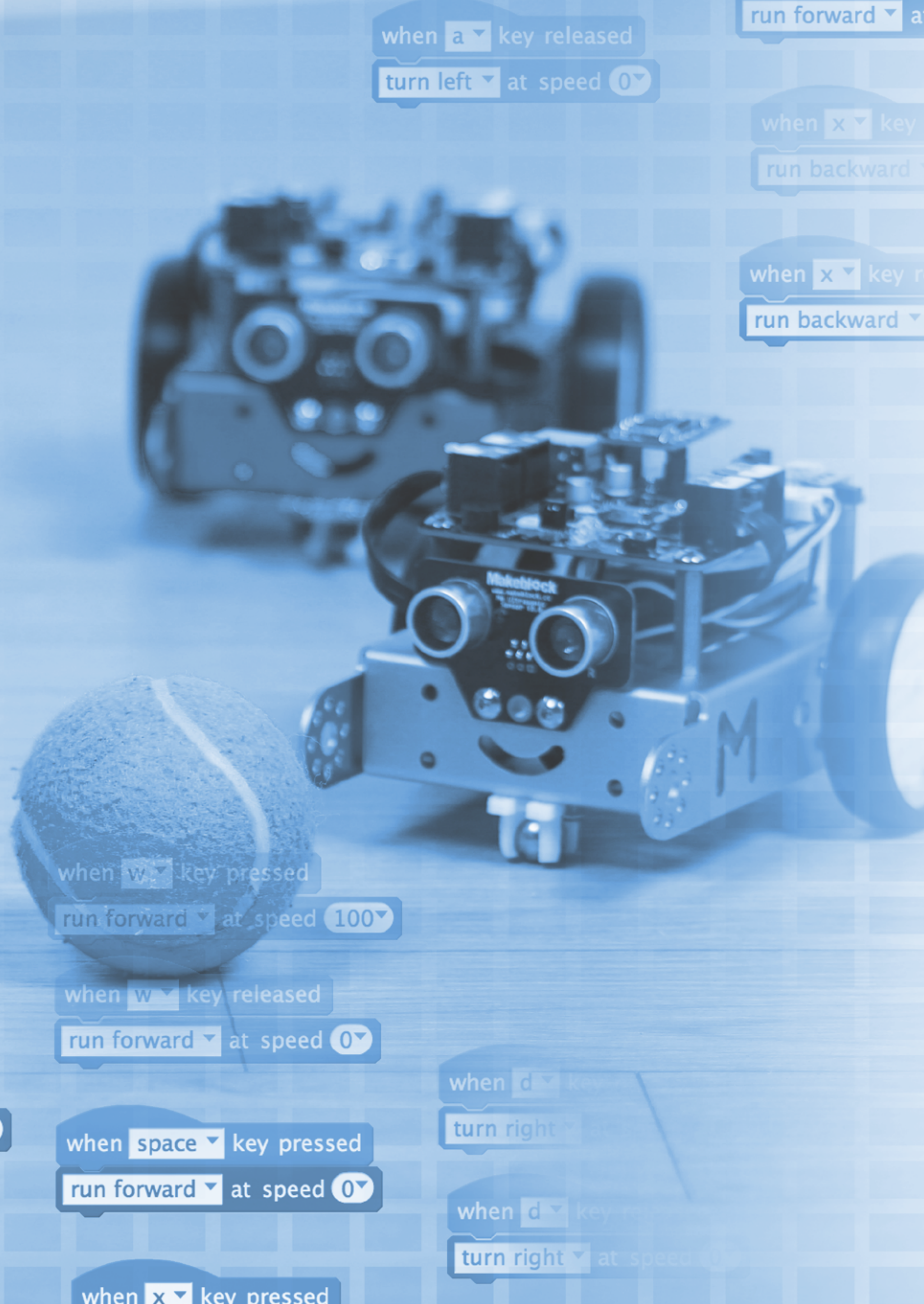
- More able children could be encouraged to add in option for setting the speed by using the numbered buttons. They will need to make a Variable in the Data&Blocks Section. Call it Speed and then set the speed based on what number you press from 0-9.



```
when space key pressed
forever
  if ir remote ↑ pressed then
    run forward at speed Speed
  else
    if ir remote ↓ pressed then
      run backward at speed Speed
    else
      if ir remote → pressed then
        turn left at speed Speed
      else
        if ir remote ← pressed then
          turn right at speed Speed
        else
          run forward at speed 0
```

```
when space key pressed
forever
  if ir remote R1 pressed then
    set Speed to 50
  if ir remote R2 pressed then
    set Speed to 60
  if ir remote R3 pressed then
    set Speed to 70
  if ir remote R4 pressed then
    set Speed to 80
  if ir remote R5 pressed then
    set Speed to 100
```

Figure 49. Variable speed remote control solution example for more able children.


when a key released
turn left at speed 0
run forward
when x key
run backward
when x key
run backward
Makeblock
M
when w key pressed
run forward at speed 100
when w key released
run forward at speed 0
when d key
turn right
when space key pressed
run forward at speed 0
when d key
turn right at speed
when x key pressed

13 ADVANCED FUNCTIONS

A newer function of the mBlock software is how it ties in with Microsoft Cognitive Service. This opens some interesting possibilities for the software moving forward.

So what is Microsoft Cognitive Services? In simple terms it is a computer that is we can use to handle more complicated processing. Instead of the computer you are sitting at processing the data itself, it will send it off via the internet to get processed and the result is sent back.

What this means is that things like voice recognition that take a lot of processing power to do well, can be used on devices that wouldn't normally be able to. It's why things like Siri, Google Voice and Cortana don't work very well when your device isn't connected to the Internet, most of the hard work is happening off the device.

By plugging into Microsoft Cognitive Service it allows the mBlock to do some things it couldn't do by itself.

Now Microsoft Cognitive Services has lots of different parts, mBlock software only uses some parts.

Using the Microsoft Cognitive Service the software can recognise what you have said, or track faces and read text. This is at the moment still in the early stages of being implemented but it does open up some interesting possibilities.

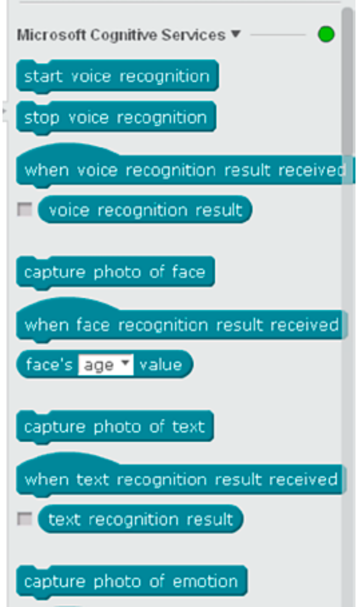


Figure 50. Image of Microsoft Cognitive Service blocks.

- You could make your mBot respond to your voice commands.
- You could allow your mBot to recognise your mood and react in different ways.
- You could control your mBot through gestures.
- You could control your mBot through command cards.

These are all still being developed at the moment, none are perfect yet for classroom use. But later books should include more information. If you are eager to explore these functions there is more information at http://www.mblock.cc/docs/guide-microsoft-cognitive-service/

Build Your Own Blocks.

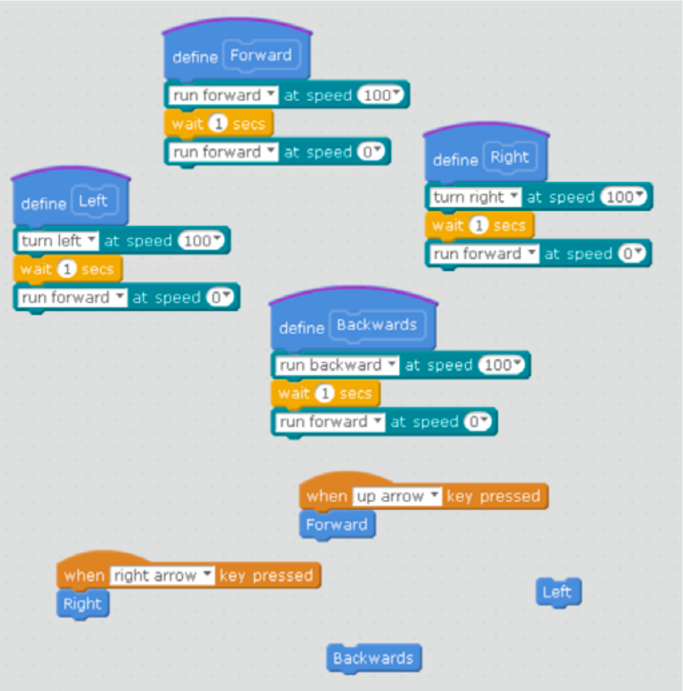
Under the Orange section on mBlock (and under its own dark purple tab section in Scratch) you have a section where you can make your own blocks. These blocks act like procedures. When you use the block it will always run the program that has been set.

For example, I could make a block that is called Stop. This block would be set up to stop all motors and end any running programs.

It is an important function as it allows you to simplify your programming, by abstracting away details to make things easier to understand. Your program will be easier to 'read'

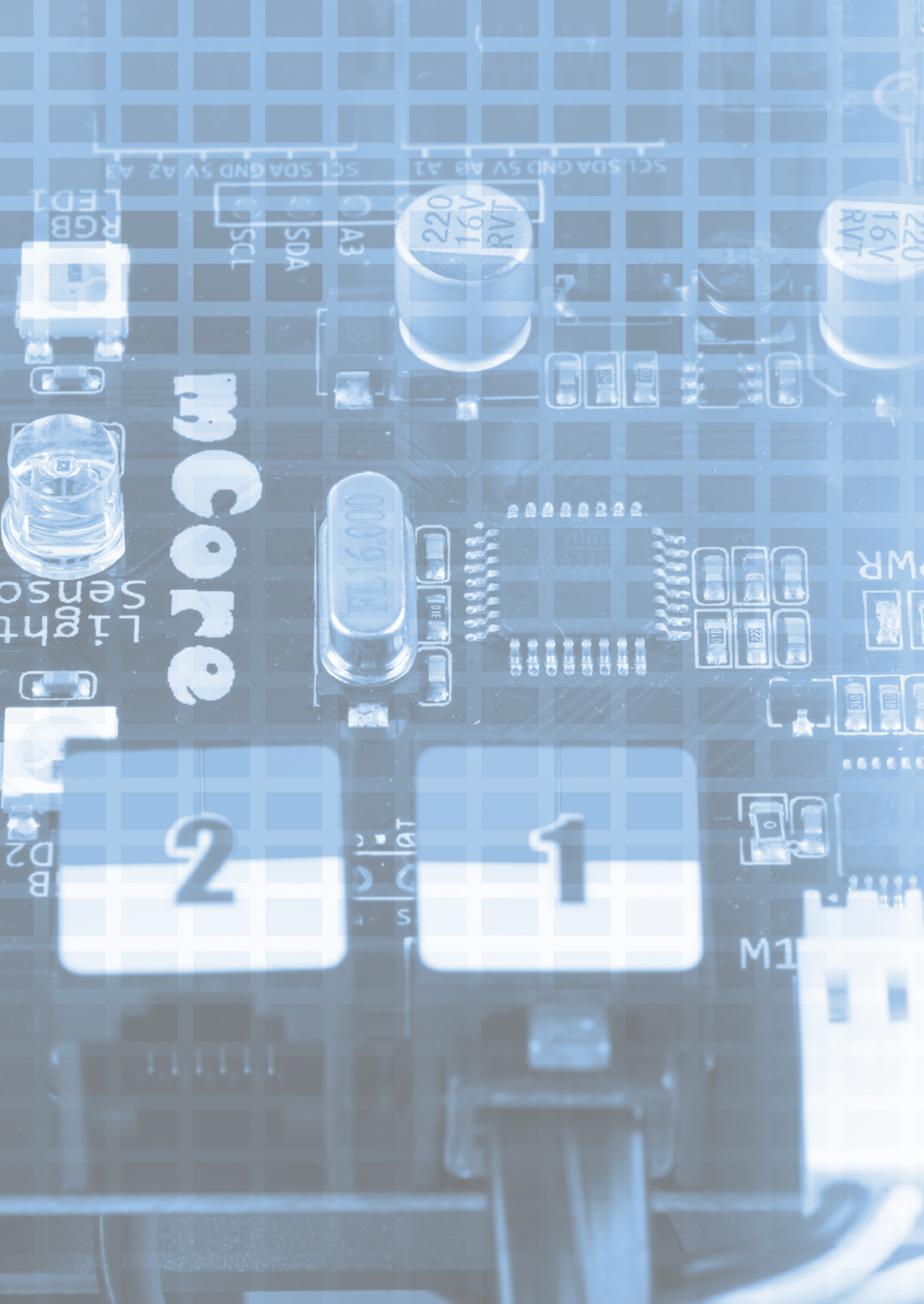
To create a block it is as simple as clicking 'Make a Block' and giving it a name, for example green. A block appears called define green and then below you add the program that you want to run when that block is used.

These could be used to create movement blocks for forward, backwards, left and right. See example below. So you could programme the mBot like more simple robots such as a TTS Beebot.



```
define Forward
run forward at speed 100
wait 1 secs
run forward at speed 0

define Right
turn right at speed 100
wait 1 secs
run forward at speed 0

define Left
turn left at speed 100
wait 1 secs
run forward at speed 0

define Backwards
run backward at speed 100
wait 1 secs
run forward at speed 0

when up arrow key pressed
Forward

when right arrow key pressed
Right

Left

Backwards
```

Figure 51- Image of example BYOB (Build your own block) program.


SCL SDA GND 5V A2 A3
SCL SDA GND 5V A0 A1
RGB LED1
SCL
SDA
A3
220 16V RVT
mCore
Light Sensor
FL16.000
PWR
2
1
M1

14 ABOUT THE CHALLENGES

The aim of these challenges is to practice and apply what's been learned in the lessons in a fun setting allowing children to experience and experiment the different skills learnt whilst developing their computational thinking.

Challenges are described with as much detail as possible while being flexible so teachers can adapt to their own class setting as appropriate according to their different learners levels, interests, time and space available.

We recommend teachers to try challenges at first with one or two robots in a quiet setting before involving the whole class if possible. This way you'll be able to identify potential barriers and learning opportunities when doing with a whole class.

In our experience, is best to keep challenges as simple as possible: most time will be spent in classroom management. If at all possible, try challenges with smaller groups or half a class; it will be more fun and beneficial for children and much easier to manage. You could attempt do a final session with the winners of each group.

All challenges follow the format explained below. Please read carefully and plan accordingly before attempting to do with a whole (or half) class, and have all materials ready for each challenge. To perform challenges in a normal lesson time can be a challenge in itself! Plan and try, the second time you attempt the challenge it will be easier.

Each Challenges will follow this format:

Heading	Explanation
Aim	What is the challenge all about?
Materials	Everything you need for the challenge.
Skills	Robot skills you need to complete the challenge.

Then each challenge will be broken into 3 parts and go into series of detailed step by step instructions.

	Explanation	Approx. Timing
Set up	Guide on how to get the challenge going in a classroom setting.	15-30 Minutes
Rules (points, penalties, time/control method)	How to run the challenge and decide who the winner is!	15-30 Minutes
Top Tips and Extension	Our best advice on how to run challenges and ways to extend challenges to push your students robot skills.	15-30 minutes

Please feel free to adapt, modify, extend challenges at your will, you know best your students and your school setting, be bold and brave and plan, plan, plan.

15 OVERVIEW OF CHALLENGES

Challenge 1

A-maze-ing: navigate through an obstacle course

Overview: In this challenge children need to navigate through a simple maze in the shortest time controlling the mBot remotely using the keyboard.

Challenge 2

Pop: robot battle, can you pop your opponent's balloon?

Overview: Each robot will be armed with a pin and have a balloon, each pair of robots will battle to try and burst the opponents' balloon with their pin.

Challenge 3

Sumo(ish) fun: push the other bot out of the sumo arena!

Overview: The aim of this challenge is to push the other mBot out of the sumo arena before the end in three rounds; best of three wins the match!

Challenge 4

Robot football

Overview: In this challenge, teams of 3 or 5 robots compete to score the most numbers of goals, winner gets through until 2 reach the Cup final.

Challenge 5

Art-bot: Drawing shapes

Overview: In this challenge, robots draw geometric shapes of equal size; create an artwork with multiple shapes. Everyone's a winner!

Challenge 6

Shake that (m)Bot dance competition

Overview: Whose got the best moves, program your robot to sing and dance to the music with its own light show.

Challenge 7

Wakey Wakey Mr mBot

Overview: Program the best waking up routine for the mBot using the light sensor, LED panel, sounds, movement and a light show. Who said waking had to be boring?

16

CHALLENGE 1: A-MAZE-ING!

Aim: Navigate a simple maze , by remote controlling the mBot, in the shortest time.

Materials: Gaffer tape, stopwatch, 2x2 square meters of space (to set up course).

Skills: You need to have completed lesson 1 before attempting this challenge. Alternatively, you can also attempt this challenge after completing lesson 7 using the remote control.

Set up:

All children (or pair of children) should have an mBot that can be controlled remotely using the computer keyboard. If no remote control program has been set up or saved, this should be done first, check Lesson 1 for complete details how to do this.

Set the rules of the challenge before hand and set a time limit.

Beware of time, if working with a full class of 30, 15 mBots (one per pair) and time limit for course of three minutes, it will take at least 45 minutes to run. We recommend setting up two (or three, even four) exact courses and running them at the same time, with the winner of each maze facing each other in a grand final or choosing the overall shortest time as the winner.

Mark your maze on the floor using the tape, below is an example maze drawing, where the side is approximately 1.5m by 2m wide:

You can try your own design, but keep it simple and test beforehand if possible. Ask children to test their code before entering the arena.

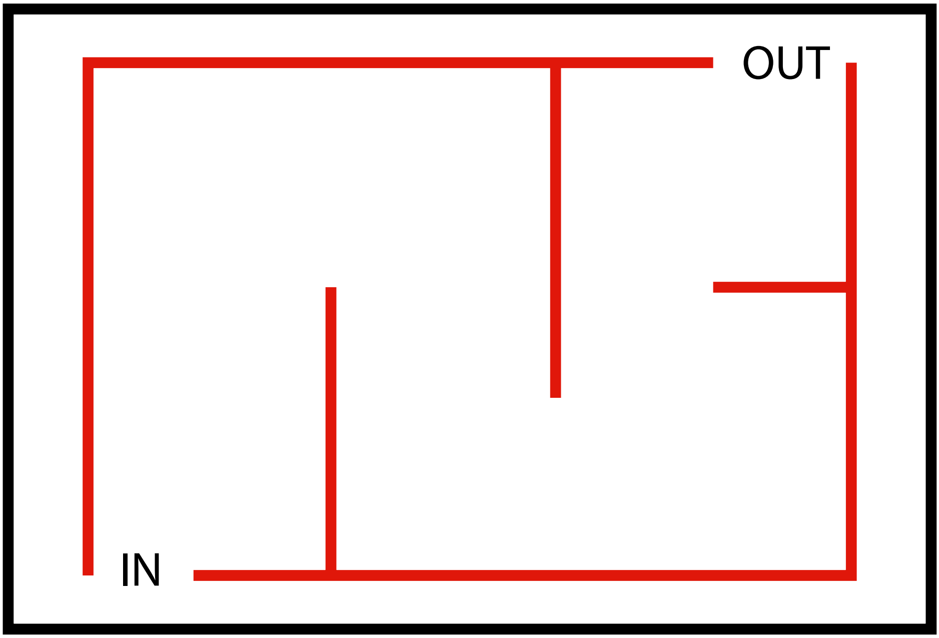


Figure 52. Example of a simple maze (tried and tested in the classroom)

Rules:

1. The robot that completes the maze in the shortest time wins. Create a table with times so others can see the time to beat.

2. Mark how far each team gets on the course in case no pair can finish the course so you can declare winner the mBot who came closest to the maze exit (or second and third place).

3. Create a rule for when robots touch the line: one line touch and back to start, three line touches and you're out.

Top Tips:

- Tell class that their success depends mainly on two things: their code and the driving controlling. Higher speed equals less control.
- If children when sitting at the computer do not have direct view of arena, children can work in pairs: one child communicates to the other where to go while the other controls the robot using the computer keyboard. This encourages teamwork and communication! Make sure the drivers (the one who controls the robot) cannot see the robot or maze so the competition is fair. Ask group to keep excitement under control and noise level down so teams can communicate.
- If enough time, try to let every pair have two goes. Encourage pairs to think how they can improve for their second run, change code? Better driving? Better communication?

Extension:

You can extend this challenge by:

- Creating a more complicated maze.
- Run an interclass competition if there is more than one class per year in your school.
- Add obstacles to the maze, you don't have to be fancy just use anything at hand in your classroom such as a box, pencil, ruler. You could even add or deduct points/seconds when touching the extra items!

when w key pressed
run forward at speed
when w key released
run forward at spee
when a key pressed
turn left at speed 100
when a key released
turn left at speed 0
when space key pre
run forward at spee
when x key presse

17

CHALLENGE 2 - POP: ROBOT BATTLE, CAN YOU POP YOUR OPPONENT'S BALLOON?

Aim: Pair of robots battle each other to try and pop the opponents balloon! Each robot will have a balloon and is armed with a pin, first one to burst the other wins!

Materials: Metal wire (garden or craft wire), pins, balloons, and tape.

Skills: You need to have completed lesson 1 before attempting this challenge. Alternatively, you can also attempt this challenge after completing lesson 7 using the remote control.

Set up:

Robots will face off against each other in a non-destructive battle that will test the driving skills of the operator and the manoeuvrability of the robot. Robots will be armed with a pin on the end of a metal wire and must use this pin to burst balloons attached to the other robot.

Test balloons and needles before hand: do the balloons burst too easily? Is the pin sharp enough? How much do you need to inflate balloons? Make sure you have plenty of spares and it might be a good idea to have inflated balloons ready.

This challenge is set in a square arena (like a boxing ring). The square can be between 1.5 and 2.5 meters square (marked with a tape). You can build or set up a short wall around the edge to prevent robots from leaving the arena.

Set up two arenas if possible so more battles can be run simultaneously.

Rules:

1. A wire will hold the balloon and a separate wire the needle.

2. The wires should be attached to front of the robot.

3. The length of the wires, once attached to the mBot should be around 10cm.

4. The balloon and needle should be attached to the top of the wire, needle at the front and the balloon at the back.

5. You can use tape, blue tack, or any other materials to attach the wire to the mBot.

6. You are only allowed to re-attach the wire, if it falls off the robot, a maximum of 3 times. The 4th time you're OUT! Test before hand to avoid disappointment.

7. Robots have to stay inside the arena, if it leaves the arena (completely) three times you're OUT.. A draw should be made at the beginning, where mBots are chosen at random to face, each other in a cup style format: winners stays for the next round, loser is out. For example, If starting with 16 robots, there will 8 battles in the first round, then a new draw is made after this round with the 8 winners until a final round. If the number of robots is less than 16 or an number even you can use this online tool: www.daftlogic.com/projects-cup-fixture-generator.htm, simply enter the number of teams and their name and a full list of fixtures will be generated for you. Max 3 minutess per battle, if no winner after 3 minutes you can chose as winner the robot who went out of the arena plus amount of times the wire had to be re-attached the least number of times. Or flip a coin to decide the winner. Remember time is at a premium. The winner is the first robot to burst the other balloon!

Top Tips

- Make sure your wire is strong before the battle begins.
- Test balloon can burst by needle.
- Keep needles protected with a tape when not in battle to avoid any injuries or accidents.
- Have two arenas at the same time, with a grand finale between the winners of each arena.
- Keep an eye on time, it should take 10-15 mins to set up robots, 45mins to 1 hour of battle time, balloons will have to be re-attached after each round, winners only! (if enough time and space you can let round losers battle each in a separate arena)
- Warning: noise level during this activity can be very high due to over excited children, think of ways how to manage behaviour using some of your usual class management tools… good luck!

Extension:

You can extend this challenge by:

- Trying with self-driving mBots! It's an interesting computing thinking exercise to investigate how to program an mBot for this kind of battle and contrast the program against a typical object avoidance self-drive program. It is unlikely you will have time to try this so why not have a class discussion at the end?

- Have battles with more than two bots at a time, even all the robots at once! (You will need a bigger arena). Warning: Mayhem may ensue!

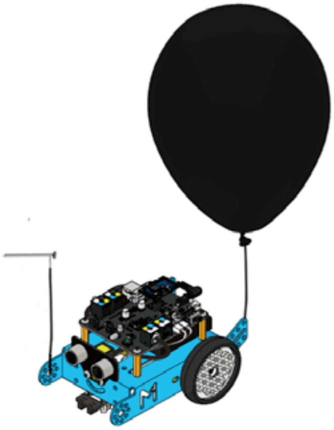
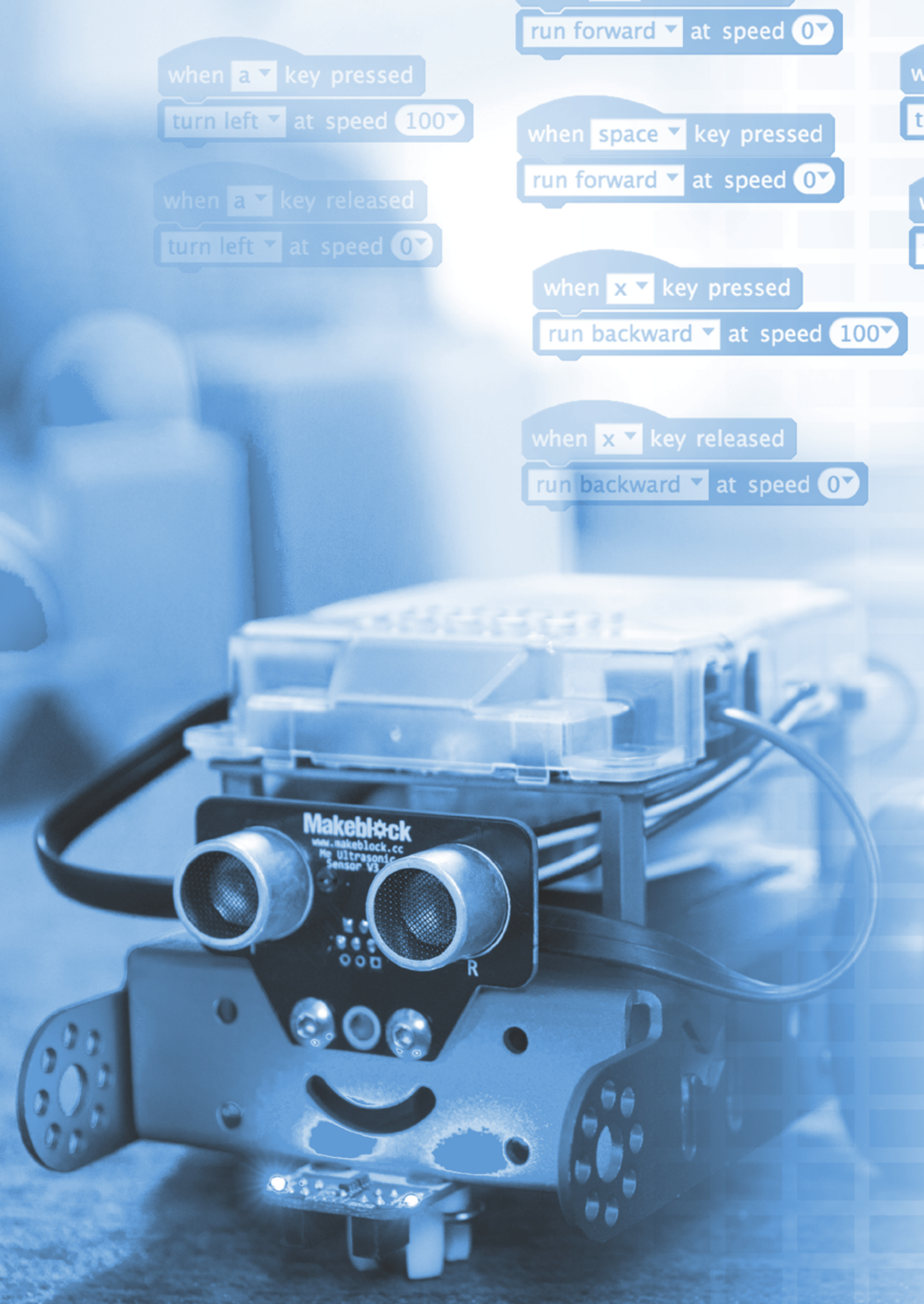


Figure 53. Diagram of mBot with balloon and needle.



```
when a key pressed
turn left at speed 100

when a key released
turn left at speed 0
```

```
run forward at speed 0

when space key pressed
run forward at speed 0

when x key pressed
run backward at speed 100

when x key released
run backward at speed 0
```

18 CHALLENGE 3: SUMO(ISH) FUN: PUSH THE OTHER BOT OUT OF THE ARENA!

Aim: Push your opponent out of the sumo arena!

Materials: Chalk & string, tape, and foam or bubble wrap.

Skills: You need to have completed lesson 1 before attempting this challenge. Alternatively, you can also attempt this challenge after completing lesson 7 using the remote control.

Set up:

This challenge is ideally set up in a circle arena with a radius between 1m and 1.5 m, in Japanese this is called the dohyo. It's easier to mark a circle with a chalk attached to a string, place one end of the string in the centre of the circle and draw a circle. If not possible to draw a circle in your class or somewhere in the school, use tape to mark a square as in previous challenge (between 1.5m and 2.5m side).

We highly recommend protecting the corners and most vulnerable parts of your mBot with a bit of foam.

Rules:

- Each match will consist of 3 rounds of 60 seconds each between 2 robots.
- If a robot leaves the arena or dohyo, the opposing robot wins the round.

- The first robot to win 2 rounds wins the match.
- If neither robot wins the round, the referee will choose the winner, based on aggression, manoeuvrability and style.
- You'll need to mark the starting lines (shikiri lines in Japanese), mark or draw two parallel lines of 1cmx10cm (width and length) centered in the ring and spaced 10 cm from each other as in diagram below.
- The winner of each match will compete against winners of other matches, until one robot is crowned the Grand Sumo Champion. You can use a draw generator as in rule 8 of previous challenge.

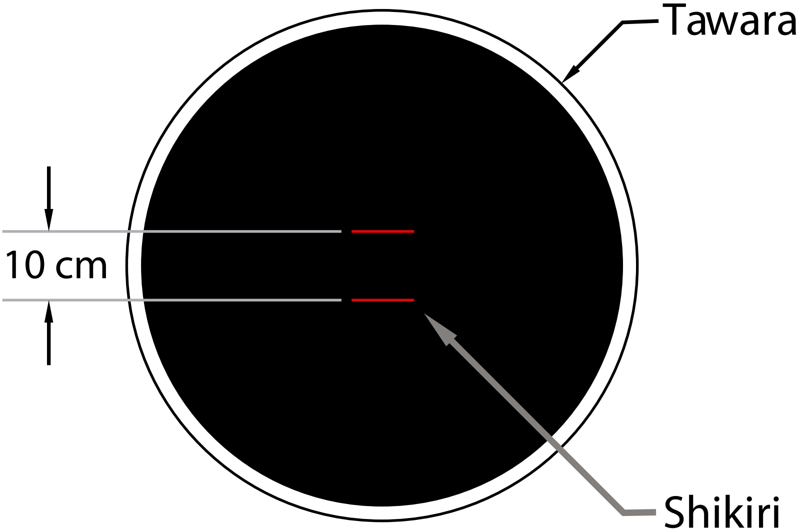


Figure 54. Sumo ring (or dohyo). A 1 to 1.5 meter radius with starting lines in red (shikiri) and border line (Tawara).

- A match will be stopped and a rematch will be started under the following conditions:

    a. The robots are locked together in such a way that no more action appears to be possible or they rotate in circles several times

    b. Both the robots are moving, but they don't appear they will ever contact each other.

c. Both robots touch the exterior of the dohyo at the same time.

d. Any other conditions under which the referee (or teacher) judges that no winner can be decided.

- Damage or destruction of other mBots is not encouraged! Robots that purposely attempt to ruin their competitors will be disqualified for the rest of the competition (teacher will referee all decisions).
- Robots are not allowed extra weapons.
- Protection: robots can be covered with foam or other protective material such as bubble wrap around the chassis to avoid long-term damage. The referee can decide if protection is adequate or not and reserves the right to ask competitors to add or remove protection.
- Before the match, contestants greet each other outside the dohyo or arena then enter the dohyo area. After that, the contestants put their robots on or behind their starting lines. No part of the robot can be placed in front of the starting line before the match begins

Top Tips:

- You could run the competition in team with points awarded per match. Team with the most points wins the competition.
- Tell students to move fast to have enough kinetic energy to push the other robot. Moving fast also makes it harder for the other bots to get you!
- Tell students to learn and accept the rules. It’s silly to get disqualified because you didn’t pay attention to the rules. If enough time, watch some sumo wrestling or robot sumo so students familiarize themselves with the sumo battle rules.

Extension:

- Try to program the mBots to run autonomously. How can you program them to find another bot and push it out of the arena? How can your mBot avoid going out before finding the other mBot?
- Why not try a rumble? The rumble is a single sumo match with up to 16 different robots to determine which robot can remain on the dohyo after

all the other robots have been pushed off the ring. The rumble could be played on a larger dohyo and the time limit could be set to 5 minutes or until there is only one robot left inside the arena. At the start, each robot must be a minimum of one robot diameter away from its nearest competitor.

Find out more about the world of the
sumo robot competitions at
http://robogames.net/rules/all-sumo.php

19

CHALLENGE 4: ROBOT FOOTBALL

Aim: Play 3 or 5 a side football with your mBots!

Materials: Tape, ball (ping pong or golf size), whistle, measuring tape, card, lollipop sticks.

Skills: You need to have completed Lesson 1 before attempting this challenge. Alternatively, you can also attempt this challenge after completing lesson 7 using the remote control.

Setup:

A 3 or 5 a side football match is played between two teams. Each team consists of three or five robots, one of which must be the goalkeeper, more robots can be added to sit ‘on the bench’. Each robot must be clearly labelled and numbered in case the referee needs to identify them during the match. The goalkeeper must be designated before the match starts. A match may not start or continue unless both teams have at least one functional robot.

- Each of the two teams has a color assigned, stick a piece of coloured card with their player number on top of the mBot.
- Control hooks: robots can be fitted with an extension to allow greater control of the ball. Attach a pair lollipop sticks to ends of the mBot front. Have a quick practice before the match. Make sure hooks are securely in place.

- Protection: we recommend protecting robots with foam or other protective material (such as bubble wrap) around the chassis, specially on corners, to avoid long term damage to mBots
- Best playing surface is hard floor, felt mat or short carpet. The playing surface must be level, flat, and hard.
- Set up of a pitch (or two). Suggested dimensions of pitch: 9m long by 6m wide for 5 a side or 4.5m long by 3m wide for 3 a side.
- Goals must be placed on the centre of each goal boundary and anchored securely to the field Surface. 1m wide for 5 a side and 0.5 m wide for 3 a side size field. You can use any static object for this.

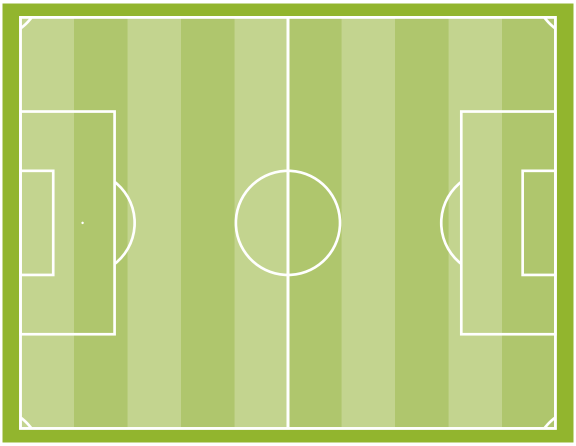


Figure 55. Image of football pitch.

The inner part of the goal can be covered with an energy absorbing material such as foam to help absorb ball impacts and lessen the speed of deflections.

- Ball: use a ping pong, golf or similar size ball.

Rules:

- The match lasts two equal periods of 10 minutes, unless otherwise mutually agreed between the referee (teacher) and the two participating teams. Any agreement to alter the periods of play (for example, to reduce each half to 7 minutes because of a limited schedule) must be made before the start of play.

- Teams are entitled to an interval at half time. The half-time interval must not exceed 2 minutes. During this time teams can adjust robots, code and discuss tactics.
- No Offside rule.
- Robots may be interchanged (with those on the 'bench'). There is no limit on the number of interchanges. Removal of robots from the game can only be made during a stoppage in play, the referee must be informed before the proposed interchange is made, a robot enters the field of play after the robot being replaced has been removed. The robot enters the field of play at the halfway line.
- Any of the robots may change places with the goalkeeper but this must notified to the referee (and other team).
- A robot can be sent off if (a) is guilty of unsporting behaviour (b) is guilty of serious and violent contact or (c) persistently infringes the Laws of the Game.
- Referee: a referee will officiate the game, if no referee is available teams have to agree decisions. The referee has full authority to enforce the Laws of the Game in connection with the match. If no referee is appointed all decisions have to be agreed between the two teams. If no decision can be agreed between teams, teacher will be called to make ultimate decision and this will be final, no TV replays or video refereeing!
- The winner if each match goes to the next in a cup style competition. Make a draw as in challenge 2.
- A coin is tossed before each match and the team which wins the toss decides which goal it will attack in the first half of the match. The other team takes the kick-off to start the match. The team that wins the toss takes the kick-off to start the second half of the match. In the second half of the match the teams change ends and attack the opposite goals.
- At the beggining of each half, each robot is located at least partially in its own half of the field. The opponents of the team taking the kick-off are at least 50cm from the ball until the ball is in play. The ball is stationary on the centre mark. The referee gives a signal or blows the

whistle for the team taking the kick-off, one robot is allowed to leave its own half of the field, as long as this robot stays within the center circle and no other robot touches the ball. The ball is in play when is ‘kicked’ and moves. The kicker does not touch the ball a second time until it has touched another robot. Game on!

Top Tips:

- “In football everything is complicated by the presence of the opposite team” (Jean-Paul Sartre) encourage teams to come up with tactics to score goals and disrupt (within the rules) the opposite team.
- The best way to gain an advantage over an opposing team is to know its strengths and weaknesses, as well as being able to predict its behavior. How can a team learn all this information? By a process known as “scouting.” Scouting consists of watching the matches of the other competitors and recording the key information about their behavior.
- Good communication between team members is essential to perform well. Get teams to choose a captain and work together as a team!
- Check the mBots screws are tight and secure before each match. Ask students to think about their code, is it better to have a fast mBot (less control) than a slower one? What is the ‘sweet spot’ or best setting between speed and control?
- Before every match, make sure that batteries are fully charged and fastened securely.

Extension:

- Play a league! You can set up a robot football lunchtime league with keen players. Keep the field marked for ease of setting up.
- Tactics are key in team sports, the more they play the better they will get at it. If your students are really enjoying playing robot football, try to find times for them to practice and play. Why not use golden time for this?

Did you know? RoboCup who organize yearly world tournaments state in their official site (http://www.robocup.org/objective) that the ultimate goal of the project is:

“By the middle of the 21st century, a team of fully autonomous humanoid robot soccer players shall win a soccer game, complying with the official rules of FIFA, against the winner of the most recent World Cup.”

when a key released
turn left at speed 0
run forward at s
when x key pr
run backward
when x key rele
run backward at
when w key pressed
run forward at speed 100
when w key released
run forward at speed 0
when d key released
turn right
when space key pressed
run forward at speed 0
when d key
turn right at spe
when x key pressed

20

CHALLENGE 5: ART-BOT: DRAWING SHAPES

This challenge is inspired by the classic robot turtle activity. Robot turtles are the oldest type of educational robots, originally designed in the late 1940s by researcher William Grey Walter and used in computer science and mechanical engineering training. Turtle robots are also closely associated with the work of Seymour Papert in computer education of the 1980s. Robot turtles are designed with pen mechanisms allowing the programmer to create a design on a large sheet of paper (https://en.wikipedia.org/wiki/Turtle_robot)). For more information and photos of Walter's and Papert's vintage robots please visit: http://roamerrobot.tumblr.com/post/23079345849/the-history-of-turtle-robots

Aim: Program the mBot to draw shapes on the floor and create robot art.

Materials: Marker pens, Tape, Paper (big format e.g. flip chart/A0).

Skills: Children should be able to code a square, rectangle, triangle and other geometric shapes in scratch. If they cannot this will need to be looked at first.

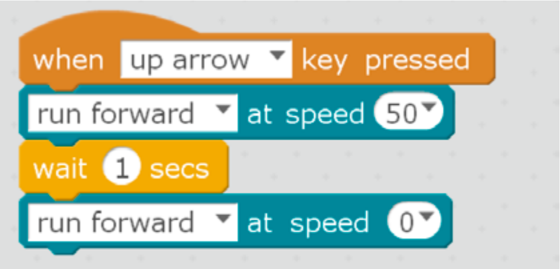
Set up:

- Attach a marker pen to your mBot and set the mBot on the floor, make sure pen is securely attach and is not too low (too much friction) or too too high (not enough pressure), you'll be able to test this when the mBot starts drawing.

- Ask children to work out how to draw a square.
- They might want to start by just drawing a line (of a set length) and making a turn (of 90 degrees)

- **To make a line of a set lenght:**

a. Try this code and use a ruler or tape measure to see how far your mBot travels.



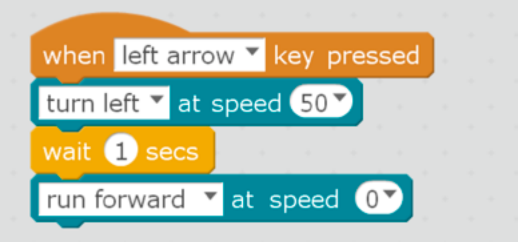
```
when up arrow ▾ key pressed
run forward ▾ at speed 50▾
wait 1 secs
run forward ▾ at speed 0▾
```

Figure 56. Example code for forward travel.

b. Now you'll have an idea of how far it will move in 1 second. Adjusting the time time will change the distance. A lot can affect this though such as, friction between pen and paper, how much battery your mBot has, efficiency of the motors, etc. Note that distance travelled could be different for each bot!

To make a turn of a set number of degrees:

a. Try this code and use a protractor to see how far your mBot turns

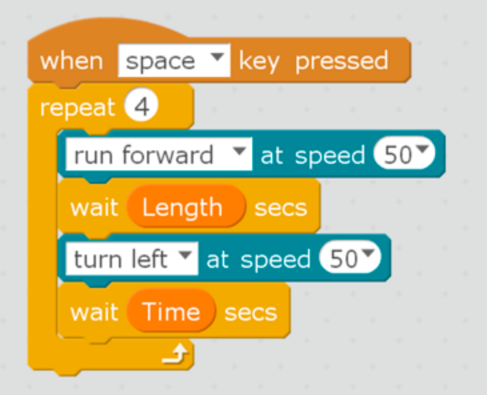


```
when left arrow ▾ key pressed
turn left ▾ at speed 50▾
wait 1 secs
run forward ▾ at speed 0▾
```

Figure 57. Example code for turning.

b. Now you’ll have an idea of how much the mBot will turn in one second. Ideally a 90 degree turn is what you want. In practice it will be a turn of somewhere been 0.5 and 1.5 seconds. A lot can affect this though such as, friction between pen and paper, how much battery your mBot has, efficiency of the motors, etc. Note that distance turned could be different for each bot!

- Now, using the information gathered above, for line length and angle of turn, the code form drawing a square should look something like the one on image below where L is the length of the desired square and T is the amount of time to turn a 90 degree angle.
- Now, think about how to draw rectangle and triangles and if you have enough time other geometric shapes, start with regular polygons! Please refer to the top tips section for instructions how to do this on screen first if you prefer.



```
when space key pressed
repeat 4
    run forward at speed 50
    wait Length secs
    turn left at speed 50
    wait Time secs
```

Figure 58. Example code for drawing a square.

Rules (Points, Penalties, time/control method):

This challenge is a little more subjective to score than previous ones, we suggest you let students get familiar with the concept of drawing shapes with the mBot first (this might take at least half the time of the lesson) and once your students have built some confidence encourage them to think more creatively to create more complex drawings using simple geometric shapes.

We propose the following format to run a competition if you wish, you could run the activity freely as well i.e. just let the children tinker and play and see what art masterpiece they come up with!

Art competition rules:

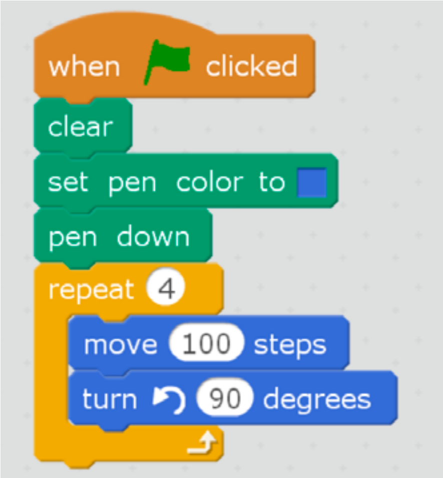
- The competition will be judged by getting students to submit what their consider their best piece.
- Each piece will then be judged in a score up to 30 based on 10 points maximum for three different categories.
- The three categories will be:
  a. Overall impression of the art
  b. Creativity and originality of the depicted theme.
  c. Quality of artistic composition and overall design.
- The winning piece will be the one with the highest score.
- A special prize can be given to the pieces with highest creativity and best composition as well.

Have fun!

Please Note:

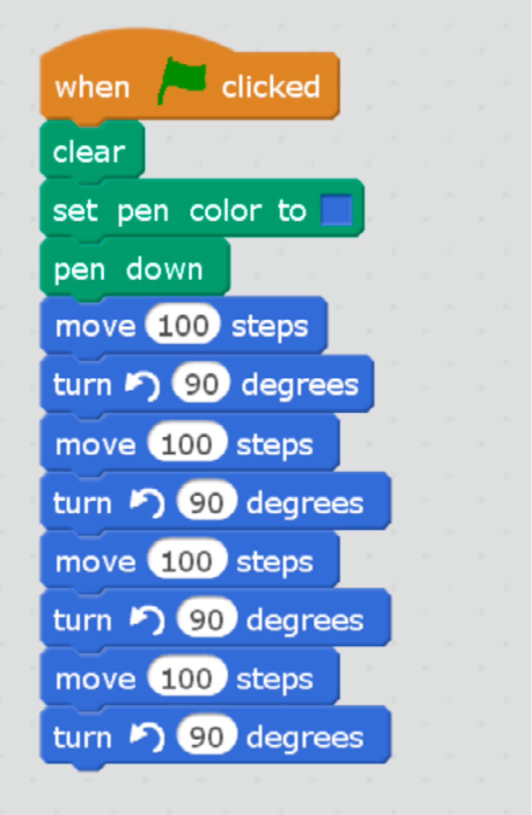
It is useful (but not necessary) to be able to code in Makeblock, or Scratch, how to draw a geometric shape first. It's very important to notice that you'll need to make a few changes to any code you have used with the screen turtle or sprite as the mBot differs quite a lot from the due to mechanical differences (e.g. battery life, assembly, motors etc). Before children can create a program for the mBot, to draw geometric shapes they need to work out how to make it travel a set distance and make an accurate turn. The mBot cannot move forward or turn a set distance due mainly to basic motors. Instead of a code with a specific line length or angle, you need to find the right amount of time to move for the mBot to be able to draw a shape. The move block needs to be substituted by the block run forward, but how far do we need it to move? We'll need to set a time limit, this will vary with each robot, so the children will have to try different settings.

If using the repeat function, it will look something like this.



```
when 🏴 clicked
clear
set pen color to 🟦
pen down
repeat (4)
    move (100) steps
    turn ↺ (90) degrees
```

Figure 59. Example code for a square using repeat function



```
when 🏴 clicked
clear
set pen color to 🟦
pen down
move (100) steps
turn ↺ (90) degrees
move (100) steps
turn ↺ (90) degrees
move (100) steps
turn ↺ (90) degrees
move (100) steps
turn ↺ (90) degrees
```

Figure 60. Example code for a square without repeat function.

Top Tips:

- Give yourself plenty of practice time to practice at your own pace before attempting this activity with a class group if at all possible. The more time you spend and understand how to create interesting shapes and patterns, the more fun you (and your group) will have.
- Try drawing screen shapes first so children get familiar with the robot turtle concept and develop their computational thinking skills. Below is an example code to draw a square. How would you code a rectangle? what about a triangle?
- Code for drawing a square on screen should look should look something like this (when not using repeat function):

Extension:

What other shapes can you make? Can you combine the shapes? Can you use more than 1 pen? Can you program the mbot to draw patterns?

What if you make a generic program that will draw any regular side shape? What inputs would you need? Do you need to set up a variable?

21

CHALLENGE 6:
SHAKE THAT BOT!
DANCE COMPETITION

Aim: Program your robot to sing and dance to music with its own choreography and light show.

Materials: Music and creativity. Assorted crafts materials.

Skills: Students need to have completed Lessons 1 and 2. Completing the art-bot challenge would be useful as this will allow children to perform more complex choreographs.

Set up:

You can perform in groups or individually. When we say individually we mean the robots, a robot might be used by two or three children at the same time.

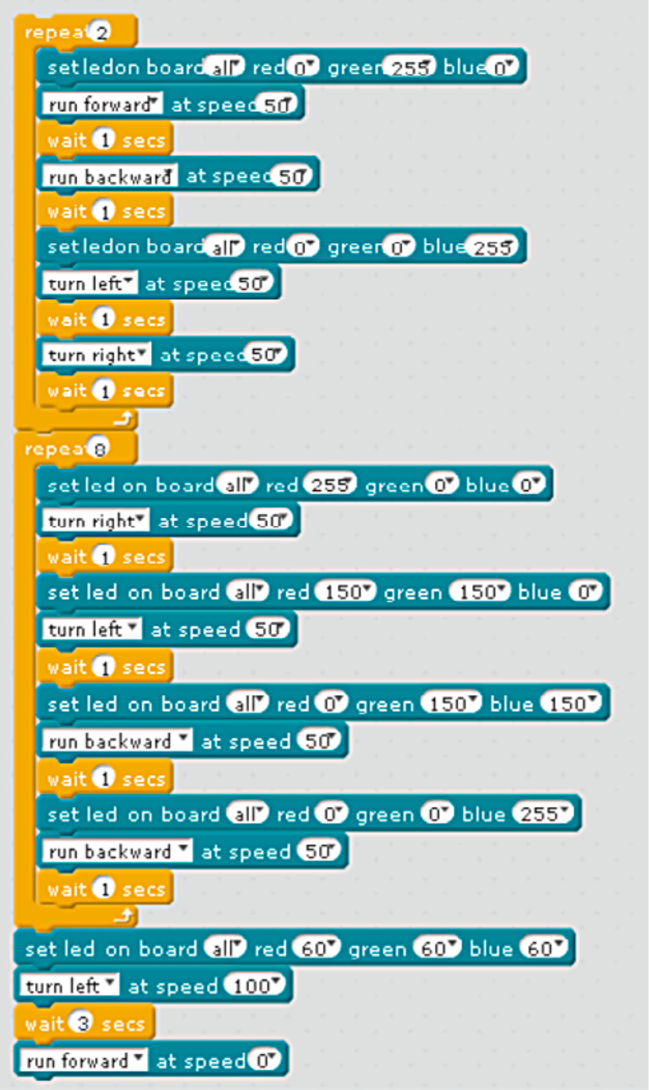
Firstly, choose a song that is upbeat and fun, with a simple structure, pop and catchy. Something like Uptown Funk by Mark Ronson should work!

You could let each robot have a different song but bear in mind the noise and complexity this will add to the challenge.

Get students to think first what kind of choreography they'd like first, let them brainstorm and plan some ideas on paper before allowing them to code. Once they start coding it gets technical and more challenging to get students to think at a higher level i.e. what is that they are trying to achieve with their code?

You can also let children decorate and dress the mBot for the occasion, let them be as creative as possible!

Once children, have planned a dance routine, get them to start coding it. A dance routine could consists if steps, a step be something like move



```
repeat 2
    set led on board all red 0 green 255 blue 0
    run forward at speed 50
    wait 1 secs
    run backward at speed 50
    wait 1 secs
    set led on board all red 0 green 0 blue 255
    turn left at speed 50
    wait 1 secs
    turn right at speed 50
    wait 1 secs
repeat 8
    set led on board all red 255 green 0 blue 0
    turn right at speed 50
    wait 1 secs
    set led on board all red 150 green 150 blue 0
    turn left at speed 50
    wait 1 secs
    set led on board all red 0 green 150 blue 150
    run backward at speed 50
    wait 1 secs
    set led on board all red 0 green 0 blue 255
    run backward at speed 50
    wait 1 secs
set led on board all red 60 green 60 blue 60
turn left at speed 100
wait 3 secs
run forward at speed 0
```

Figure 61. Dance routine example.

forward half a second, move backwards half a second, turn around, move to the left, move to the right. You can then put together a routine by combining steps. An easy way to think about the whole routine is to have a step for the intro, one for the verse of the song, one for the chorus and one step for the end (a twirl is always a good way to end!).

Remind students to add lights to their moves, one way would be to have distinctive light for each step in a routine, this could be static or dynamic. For instance, the intro could be red, first verse blue and flashing white for the chorus! You can even add notes to emphasize changes between parts, try to match the tunes tone!

Rules:

Similarly to the previous challenge, this challenge is more subjective to judge and similarly we can use the same criteria as in the artbot challenge.

We propose the following format to run a competition if you wish, you could run the activity freely as well i.e. just let the children tinker and play and see what dance routine they come up with!

Dance competition rules:

- The competition will be judged by getting students to perform their routine to the chosen song. The routine should not more than 3 seconds than the song.

- Each routine will then be judged in a score up to 30 based on 10 maximum points for three different categories.

- The three categories will be:
  a) Overall impression of the dance routine.
  b) Creativity and originality of the dance.
  c) Quality of choreographic composition.

- The winning piece will be the one with the highest score.

- A special prize can be given to the pieces with highest creativity and best composition as well.

Top Tips:

- Allow plenty of time for children to plan their routine and then implement it. They might realise that some dance moves require complex programming, they can revise their plan and adapt it to make sure they complete the challenge in the allocated time.
- Encourage children to be creative and inventive, watch some dance choreographies videos of your choice at the beginning of the session for inspiration.
- Check that you'll be able to play the video in your classroom computer beforehand, for instance check that youtube content is not blocked by your school system in advance.
- Get groups of children to try how best to synchronize their mBot start with the song, this could be done manually by starting the code and song at the same time, the easiest way to done this would be to program the robot to start the routine when the space bar is pressed, load the song on youtube (or other similar platform) and start the song with the space bar. Voila! Instant synchronicity!
- Get a disco ball for extra disco glitz!

Extension:

One way to extend this challenge would be to group robots in teams and get them to perform a song as a group, not all robots in a group have to perform the exact same routine but a level of similarity and synchronous dance will be expected.

You could let children to choose their own song instead having the same song for all, the song will have to be approved by teacher (check video is age appropriate).

22

CHALLENGE 7: WAKEY WAKEY MR. MBOT

Aim: Program different behaviours in your mBot depending on light levels using the on board light sensor and LED matrix.

Materials: Large room ideally with dimmable lights and/or blackout curtains.

Skills: Students need to have completed Lesson 3 (Page 45). Completing the dance challenge previous to this one would be useful too as it will allow children to program more complex behaviours.

Set up:

This is more of a programming challenge and most of the activity will be spend programming and testing the code. Note that, in this challenge we use the mBot on board light sensor, so for this you will need to be in a room where you can control the light level either by turning lights off or dimming them or by closing curtains or a combination of both. You'll need plenty of space for all mBots as part of the fun will be to have all robots performing at the same time.

Step 1: Install LED panel as indicated in Lesson 3 (Page 45).

Step 2: Create 'emotions' using LED panel. For instance, create a 'face' fo when robot is 'sleeping' i.e. when there is no or little light and another 'face' for when robot is awake. For example, when your robot is asleep you could try some sleepy eyes using the *show drawing* block:

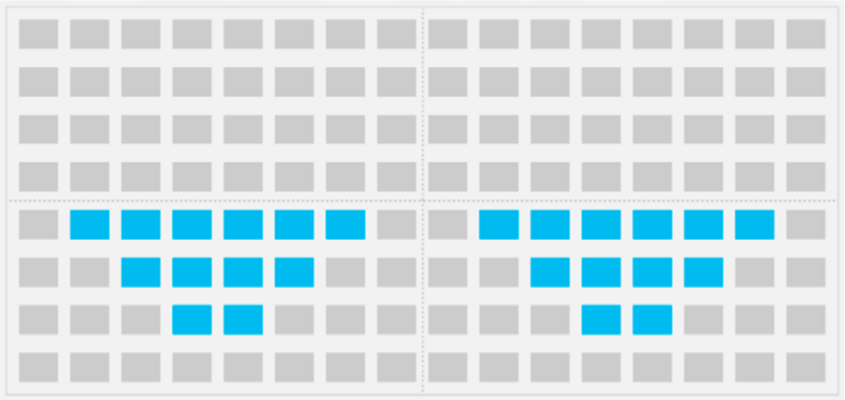


Figure 62. Example of mBot eyes closed or 'sleeping' eyes.

You'll also need an awake face also using the *show drawing* block, for instance:

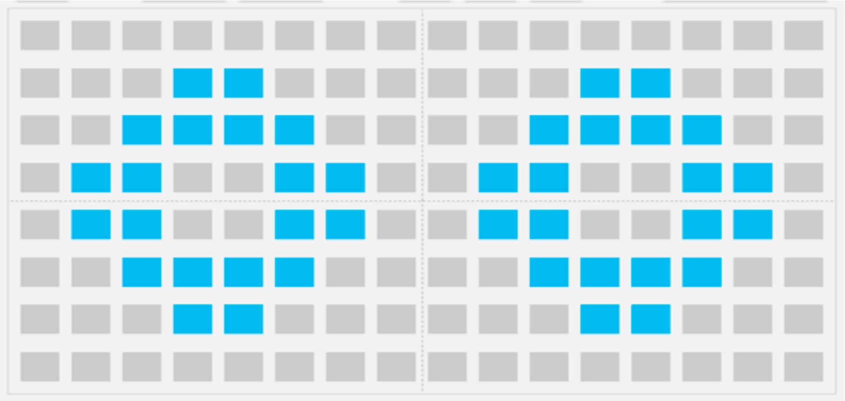


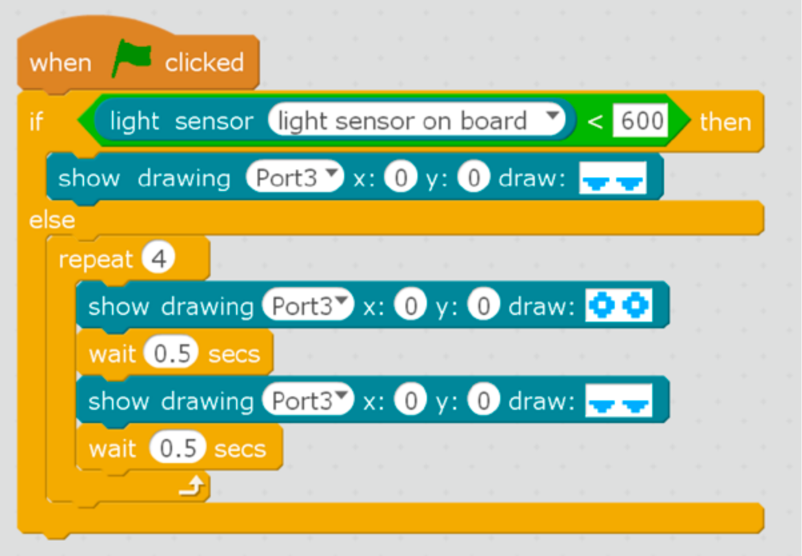
Figure 63. Example of mBot opened eyes.

Step 3: Using the light sensor.

Now we'll use the on board light sensor as an 'If then else' block (from the Control block group). The main idea here is to use a conditional block to specify two types of behaviours depending on the light level. One behaviour for low light level, is the 'sleeping' behaviour which you can keep quite simple. While the other behaviour, for higher light levels, will be more complex by combining different ways of making the mBot move, make sounds, change its 'face' and using the on board LED's.

Note that you will need to adjust the value of the light level in the if block depending on your own setting. The light sensor on board has values from 0 (no light) to 1023 (full light) so it's a good idea to start with a value round the middle, for example 600.

An example code, using the faces above, a value of 600 for the change of behaviour and getting the mBot to 'blink' once it wakes up, would look like this:



```
when ⚑ clicked
if light sensor (light sensor on board ▾) < 600 then
    show drawing (Port3 ▾) x: 0 y: 0 draw:
else
    repeat 4
        show drawing (Port3 ▾) x: 0 y: 0 draw:
        wait 0.5 secs
        show drawing (Port3 ▾) x: 0 y: 0 draw:
        wait 0.5 secs
```

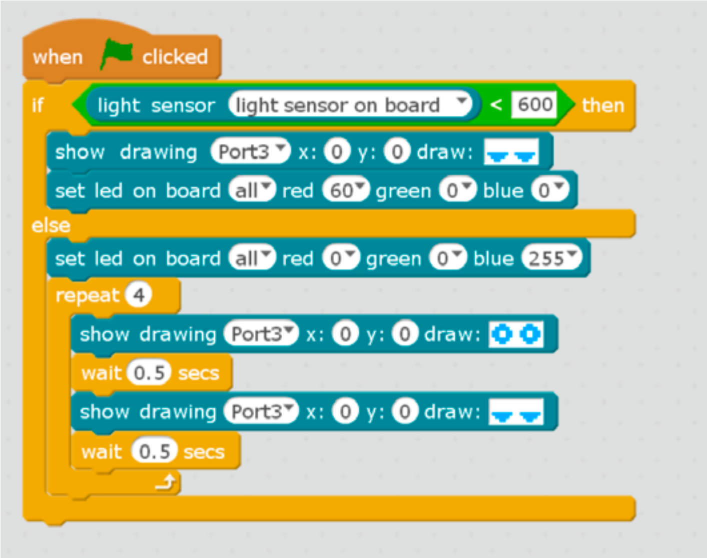
Image 64. Example code for programming 'sleeping' while lights are off (or light less than 600) and 'blinking' when then lights are on (or over a value of 600)

How many times does it blink? Is the time between open and closed eyes too long, too short or about right? How can we make it more interesting? (some possible answers to these questions are fund in the next steps).

Step 4: Programming more complex 'behaviour'.

Now, we can build on the code above to build a waking up routine. We can add some lights for when we are awake and also some dimmed lights for while we are asleep, be careful when programming the on board LED lights while asleep, if there are too bright they might trigger the on board light sensor and 'wake' the robot up.

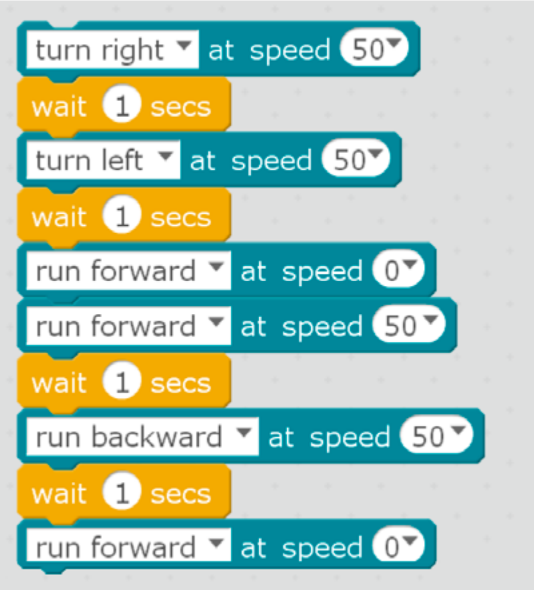
Just adding some code for the onboard LED lights would look something like:



```
when ⚑ clicked
if light sensor (light sensor on board) < 600 then
    show drawing Port3 x: 0 y: 0 draw:
    set led on board all red 60 green 0 blue 0
else
    set led on board all red 0 green 0 blue 255
    repeat 4
        show drawing Port3 x: 0 y: 0 draw:
        wait 0.5 secs
        show drawing Port3 x: 0 y: 0 draw:
        wait 0.5 secs
```

Image 65. Using the on board LED to add the ‘sleeping’ and ‘awake’ estates.

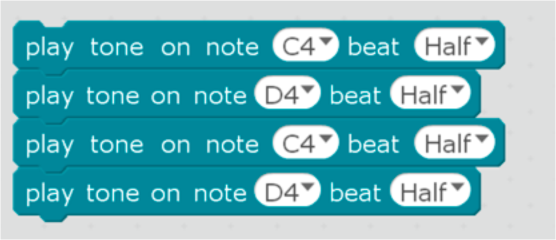
We can then add more behaviour, for instance, lets add some movement after blinking, such as moving left, then right, forward and back. Such code would look like this:



```
turn right at speed 50
wait 1 secs
turn left at speed 50
wait 1 secs
run forward at speed 0
run forward at speed 50
wait 1 secs
run backward at speed 50
wait 1 secs
run forward at speed 0
```

Image 66. ‘Stretching’ behaviour after waking up. Note that we have added a run forward at speed 0 at the end of the code to make robot stop.

Now, we can add some sounds, for instance:



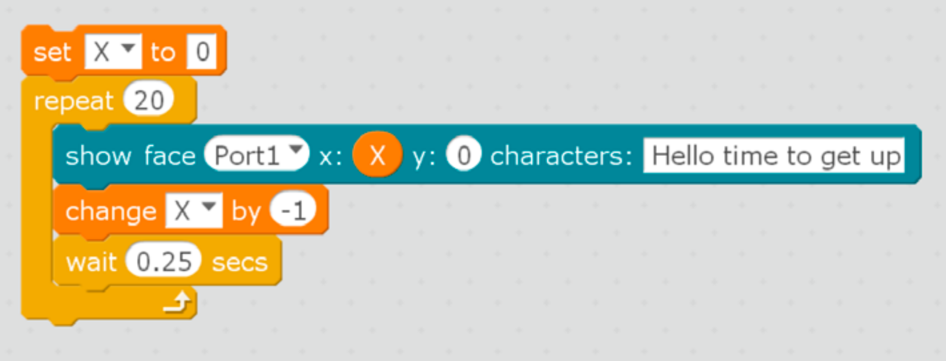
```
play tone on note C4 beat Half
play tone on note D4 beat Half
play tone on note C4 beat Half
play tone on note D4 beat Half
```

Image 67. Example of sounds using the play tone block.

Finally, in Lesson 3 we mentioned that the LED screen can only display a short text and it does not scroll. However, we can make text scroll in the LED panel by using some coding!

Note that the show face block has x and y coordinates and that they are set to 0 and 0 as default. If we make the x value change after a set time, we can make text scroll. In order to do this we need to create a new variable. Click on the orange 'Data & Blocks' menu , then select 'Make a Variable' and call it 'X'. This will be the position of the text, to make the text scroll to the left we need to change the position by -1 and wait a short period of time (for example 0.2 secs) and repeat until all text is displayed.

An example code to display a message such as: "Hello time to get up" would look like:



```
set X to 0
repeat 20
    show face Port1 x: X y: 0 characters: Hello time to get up
    change X by -1
    wait 0.25 secs
```

Image 68. Example code for scrolling text. Note that you'll need to create a new variable 'X' if your text is longer than 3 characters.

So, now you can put all of the above code, or your own, in order in the else section of the If else block. Be careful with the order and aware that you might need to add pauses (wait blocks) between certain parts. Test your code first without the conditional to make it easier to debug before putting it all together.

Once all students have finished and tested their code, turn the robots off, turn the lights off, turn the robots back on, press the green flag on the mBlock software (or whatever key you have used to activate your code) and watch!

Rules:

Similarly to the art-bot and dance challenges, this challenge is subjective to judge and we can use the same criteria as before or just simple let the children have fun and do it as a group challenge. Get all mBots ready, turn the lights on (or off) and have fun!

Top Tips:

- Allow plenty of time for children to plan their waking up routine and then to implement it as before. They might want to use some previous code and add to it.

- Try different values for the light sensor and find one that works best on your context. Start with 600 and if too low and the robot only shows its 'awake' behaviour, make it higher. Conversely, if the robot stays in its 'sleeping' behaviour try a higher number and see what happens.

- Be careful when adding the on board LED lights in the sleeping part of the code, if too bright they might trigger the on board light sensor. Use a small value, remember that values go from 0 to 255.

- Remember to use Port 3 in the show face and other blocks that use the LED matrix.

- Get students to create their routine in secret and do reveal all at the same time when turning the lights on.

Extension:

You can set a more detailed design brief for the sleeping and awake behaviours. For instance, can you make a snoring noise while asleep? Or set a different routine in which mBot wakes up, blinks first then moves, then show a smiley face, then make some noise while moving and turning lights off.

Try adding the LED matrix without taking out the ultrasonic distance sensor, in that case connect the Matrix to Port 4 and make sure you use that port on the corresponding blocks (show face, show time, show drawing). Can you code a distance measuring display?

You can program the opposite behaviour i.e. a simple behaviour for when the lights are on and a more complex behaviour for when the lights are off (or below a certain level), so instead of going to sleep the mBot does a crazy dance when the lights go off.

You could ask students to create different emotions for the robot once it wakes up using the show drawing block and add colours (using on board LEDs) to each emotion.

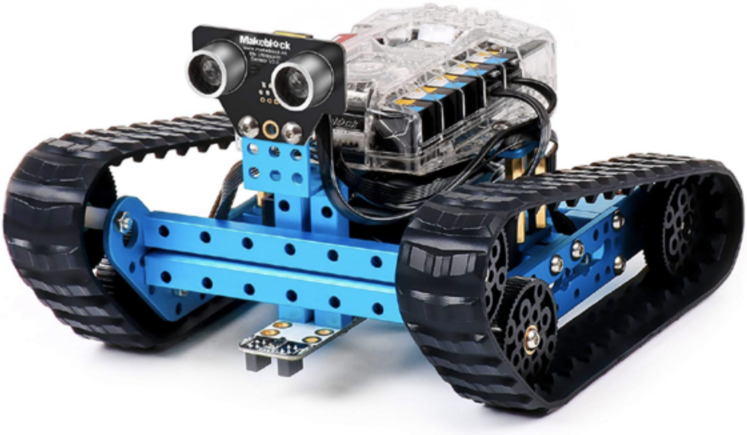
Try using the LED matrix and the line following infrared sensor. Can you program it so you know when zero, one or the two sensors are on to display something accordingly?

23 OTHER MAKEBLOCK PRODUCTS

The mBot is only one of many products that Makeblock offer.

mBot itself is just the first level of a series of customisable robot kits. They also offer:

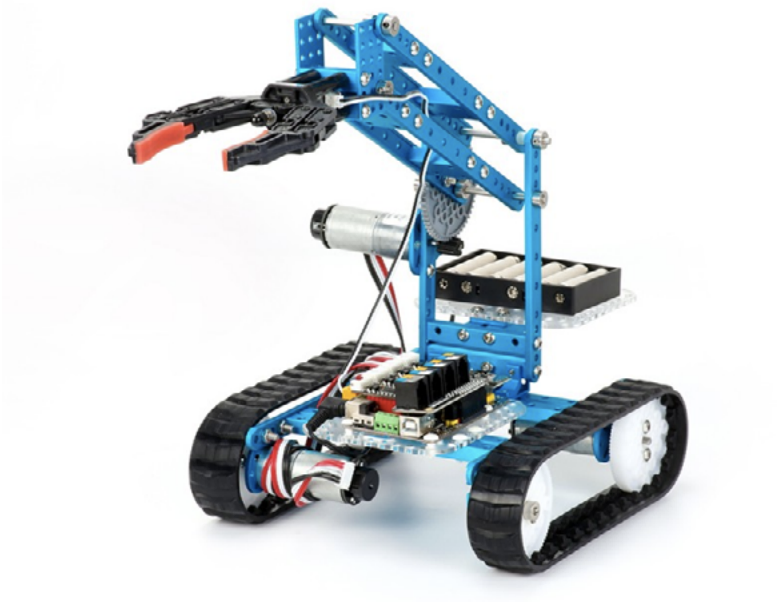
mBot Ranger - https://www.makeblock.com/steam-kits/mbot-ranger



The kit can be used to create 3 different styles of robot, the control board also has more built in sensors and a capacity to add more input and output devices. It is programmable on a computer or tablet and supports block based programming.

Age Range: KS3 and above.

Ultimate 2.0 - https://www.makeblock.com/steam-kits/mbot-ultimate



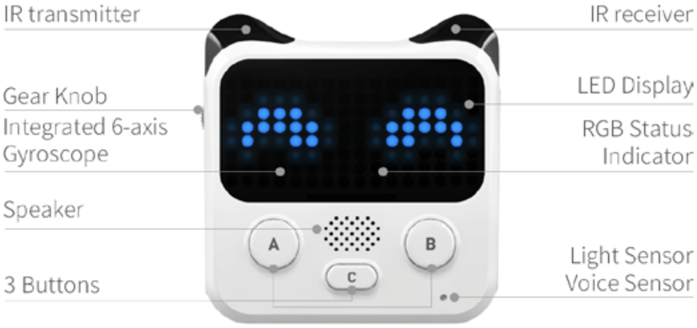
The kit comes with instructions for 10 different designs and can be used to build more personalized creations. The control board is more advanced again and can be used to drive a wide range of servos and motors. It is also compatible with Raspberry Pi and Arduino. It is programmable on a computer or tablet and supports visual or block based programming, Python, Arduino IDE and other programming languages.

Age Range: KS4 and above.

They also produce the following:

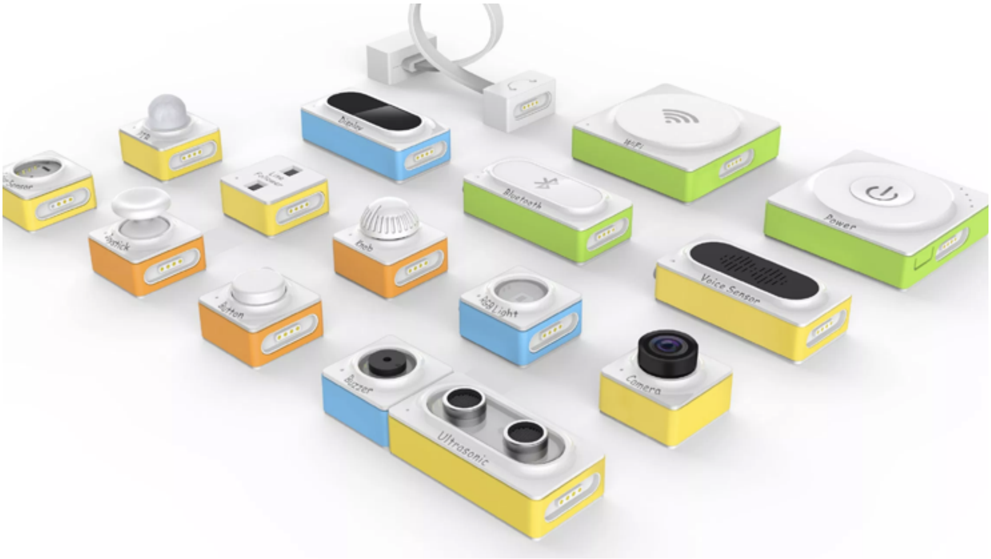
CodeyRocky - https://www.makeblock.com/steam-kits/codey-rocky

Is their newest robot that combines a small handheld 'brain' with a tracked body. The handheld section has a LED screen and a range of sensors. Then it is plugged into the tracked body it gains a extra sensor and the ability to move. Think of the brain like an advanced micro:bit. It is programmable on a computer or tablet and supports both block based programming and Python.



Age Range: KS2

- https://www.makeblock.com/steam-kits/neuron



Is a series of blocks that can be joined together via magnets and then programmed using a computer or tablet. You have a range of different input and output blocks that will allow you to build all wide range of contractions. If you have ever seen littleBits or SAM Labs it is in the same vein. It is programmable on a computer or tablet and uses a block based programming langage.

Age Range - KS2 and above.

Airblock - https://www.makeblock.com/steam-kits/airblock

The airblock is a programmable drone, it is modular meaning that it is quite robust as parts can be moved around. The extra bonus with this is that it can be reconfigured into a hovercraft. It can only be used with a tablet and is programed using a block based programming languge.

Age Range - KS2 and above.

24. Bibliography

Oxforddictionaries.com, (2017) ‘Definition of robot in English’, [online] Available from: https://en.oxforddictionaries.com/definition/robot (Accessed 11/6/17).

Image of Curiosity Rover, Image credit: NASA/JPL-Caltech/MSSS, accessed August 2017, <https://solarsystem.nasa.gov/images/content/PIA16763_720.jpg>

https://en.wikipedia.org/wiki/Turtle_(robot)