UCL

The Scholars Programme

Taming the BristleBOT

Key Stage 3 Programme

IEP Integrated Engineering Programme

Pupil Name

PhD Tutor

UCL Engineering Faculty Vision – Our Integrated Engineering Approach

"To change the world, you need to be taught differently!"

Have you ever wondered what engineers actually do? Engineers design, create, explore, innovate and work collaboratively with experts from many other disciplines for the design and development of tools, materials and technologies in medicine, communications, construction, environmental sustainability, food technologies, space programs, transportation, sport and entertainment. From artificial bones and smart medical machines that perform delicate surgery, to cutting edge sports training equipment and high-tech clothing, to everyday household appliances, personal devices and energy-efficient equipment, engineers help shape the future and make a difference in the world we are living in.

Our changing world is in need of engineers who are different to those of past generations. The future needs a new kind of engineer, the kind of engineers who can: work with all different kinds of people; think creatively and critically; ask thought provoking, smart and radical questions; solve complicated problems; communicate clearly and confidently; work well in teams; and most certainly, design and make innovative and sustainable solutions. We don't know what the future will look like, but we hope our engineering graduates will play a part in shaping it for people all over the world.

At UCL, we understand that this is a very big ask. Our Integrated Engineering undergraduate programme (IEP) is designed to include new and exciting ways of teaching so that all of our students get the chance to develop and practice their own skills while learning all about their technical subjects. The opportunities our students have to work on real world problems gives them insight into the type of work professional engineers really do and how the problems they solve and the solutions they come up with can change lives.

One of the first things all of our IEP students come to understand very early in their studies here at UCL is that there isn't one right answer to engineering problems. This might sound frightening, but actually this is one of the best parts about being an engineer! What this means is that every engineer has the opportunity to explore her or his own ideas, ask questions that will change how you look at the problem that needs to be solve and work with others to create new and exciting innovations. It also means, that the choices made by engineers will have consequences (both good and bad) and it is important to know how to deal with them.

There is a process that all engineers work through to determine how to build something whether that be a skyscraper, a bicycle, a music player, an electric engine, new plastics, sweet treats and ice cream, medicines and vaccines or mobile apps. The process is called the engineering design process. Through the IEP, our students have many opportunities to explore all of the phases throughout this process. It is a cycle that often requires the engineers to try

something and then try again after learning something new. From the start, engineers gather information and conduct research to understand the initial problem or need for the project to be addressed. This means investigating who is involved at the start of the project and who will be affected by the solution. Engineers need to understand the people who are affected by the original problem so that they can design a solution with their concerns and needs in mind. Then they imagine and brainstorm ideas for the solution – at this point, no idea is a bad idea and it is really good to think of wild ideas too. Engineers work together in teams so that they can generate as many ideas as possible, so they have lots of choices to work with. Together they select the most promising idea and embark upon a design that will require them to create drawings, make precise decisions on the materials and construction, as well as manufacturing and fabrication technologies to use. Whilst making these decisions, engineers go through a time of testing by creating and testing many prototypes. This phase of the design process often comes with lots of setbacks and failures, but the testing team works together to learn from every test to make improvements until the design is good enough to meet the needs and requirements of the problem they started with. After the testing and prototyping stage, the teamwork together to manufacture the real thing. The design process doesn't end there though as one of the most important parts of the whole process is to understand if the solution is accepted and used as you intended. The end of the design process always gives the engineer a time to reflect and learn so that their next project is benefited by what they experienced and learned in each of their previous design projects.

The Integrated Engineering Programme is intended to give your students an opportunity to put their learning into practice by working in an interdisciplinary, problem-based learning, industry linked and design focused environment. At its core, is the deliberate attempt to make use of and explore the creative and stimulating aspects of design as practiced by 'real' engineers and computer scientists in industry and the professional skills needed to be successful in the enticing and highly competitive working world.

With thanks to

This programme has been designed through a collaboration between The Brilliant Club and UCL Engineering Faculty. With special thanks to the following people:

- Dr Benn Thomsen, Reader, UCL
- Natcha Sujaritworakun, MEng Student, UCL
- Sarah Dreischer, MEng Student, UCL
- Ben Dolman – PhD, University of Manchester
- Emanuela Tilley - *Director, Integrated Engineering Programme at UCL Engineering*

Timetable and Assignment Submission

Timetable – Tutorials

Tutorial	Date	Time	Location
1 (Launch Trip)			
2			
3			
4			
5			
6 (Feedback)			

Timetable – Homework Assignments

Homework Assignment	Description	Due Date
Tutorial 1		
Tutorial 2		
Tutorial 3		
Tutorial 4		
Tutorial 5		

Assignment Submission – Lateness and Plagiarism

Lateness	
Submission after midnight on _____	10 marks deducted
Plagiarism	
Some plagiarism	10 marks deducted
Moderate plagiarism	20 marks deducted
Extreme plagiarism	Automatic fail

The Brilliant Club KS3 Programme – Pupil Feedback Report

Grade	Marks	What this means
1st	70+	Performing to an excellent standard at GCSE
2:1	60-69	Performing to a good standard at GCSE
2:2	50-59	Performing to an excellent standard at GCSE
3rd	40-49	Performing to a good standard at Y7/8
Working towards a pass	0-39	Performing below a good standard at Y7/8
Did not submit	DNS	No assignment received by The Brilliant Club

Lateness	
Any lateness	10 marks deducted
Plagiarism	
Some plagiarism	10 marks deducted
Moderate plagiarism	20 marks deducted
Extreme plagiarism	Automatic fail

Name of PhD Tutor			
Title of Assignment			
Name of Pupil			
Name of School			
ORIGINAL MARK / 100		FINAL MARK / 100	
DEDUCTED MARKS		FINAL GRADE	

If marks have been deducted (e.g. late submission, plagiarism) the PhD tutor should give an explanation in this section:

Learning Feedback Comment 1 - *Enter Key Learning Priority Here*	
What you did in relation to this Key Learning Priority *Enter feedback here*	How you could improve in the future *Enter feedback here*
Learning Feedback Comment 2 – *Enter Key Learning Priority Here*	
What you did in relation to this Key Learning Priority *Enter feedback here*	How you could improve in the future *Enter feedback here*
Learning Feedback Comment 3 – *Enter Key Learning Priority Here*	
What you did in relation to this Key Learning Priority *Enter feedback here*	How you could improve in the future *Enter feedback here*
Resilience Comment	
How you showed learning resilience during the course *Enter feedback here*	How you could build learning resilience in the future *Enter feedback here*

Contents

Course Rationale

Engineers solve problems by creating new products and processes. Real world problems are often complex and this means there usually isn't one true or "perfect" answer. Making decisions is often the hardest thing an engineer must do and often not all the information is available. To overcome this, engineers take existing technologies and approaches, new and old scientific discoveries, mix them together, add new ideas and create things the world has never seen before.

During this programme you will work in small teams to build a BristleBot - a small, wild robot. Your challenge as engineers is to tame your BristleBot. To do this you will need to plan and implement modifications to enable you not only to be able to control where your BristleBot goes, but to enable it to 'see' and therefore to steer around any obstacles in its path. While working with your BristleBot you will use the same creative processes that engineers use to help them to solve all kinds of problems, both big and small. Having raced your modified BristleBot against other teams, you will be required to write a scientific report explaining and justifying the modifications you have made to your Bristle and how these impacted on the performance of the BristleBot in avoiding objects.

Throughout this programme you will learn new scientific knowledge and apply this to the modifications you will choose to make to your BrisleBot such as computer programming, electronics and turning forces. You will be working like engineers in small design groups to implement the engineering design process to create the best possible BristleBot for the final challenge.

Course Summary

Tutorial 1: What is a BristleBot? – Creating a simple BristleBot

Tutorial 2: How can we control the BristleBot? – Introduction to the electronics of a dual-BristleBot

Tutorial 3: Are BristleBots always chaotic? – Programming your BristleBot to travel in a straight line

Tutorial 4: How can BristleBots see? – Programming your BristleBot to avoid objects

Tutorial 5: What are the best BristleBot Modifications? – Challenge: moving BristleBots round a maze

Tutorial 6: Final assignment feedback

Mark Scheme Table

Skill	1st	2.2
Creativity ('Ideas')	• A high level of initiative and creativity was shown in the modifications and experiments. • A high level of initiative and creativity is shown in the presentation and content of the written work. • Ideas go beyond those introduced by the tutor.	• There is some evidence of initiative and creativity in the modifications or the experiments. • There is some evidence of initiative and creativity in the presentation or content of written work. • Ideas are based largely on those introduced by the tutor.
Practical Work ('Do')	• Experiments were very well planned. • All necessary preparation (including sourcing additional materials) was completed prior to tutorials. • Experiments were conducted in a structured manner. • Results were recorded in a logical way.	• Some aspects of experiments required more planning. • Some preparation was completed prior to tutorials. • Experiments needed to be better structured. • Some results were recorded in an illogical way.
Critical Thinking and Evaluating ('Improve')	• Clear improvements have been made to the work following feedback. • Any limitations of the experiments have been discussed. • Ideas for further improving the experiments and BristleBots are fully discussed.	• Few improvements have been made to the work following feedback. • Few limitations of the experiments have been discussed. • Some ideas for further improving the experiments or BristleBots are mentioned.
Communication	• Language is concise, clear and engaging. • Writing is well structured using the EDP. • Appropriate technical vocabulary is confidently and correctly used. • Work produced could be easily recreated by another to achieve the same results.	• Language is mostly clear but missing some detail. • Some attempt to structure writing using the EDP has been made. • Some attempt is made to use technical vocabulary. • Another could attempt to follow work produced to achieve similar results, but with some difficulty.
Teamwork	• Ideas were presented to others in a confident, engaging manner. • Others' ideas were respected and considered. • Effective teamwork led to an improvement in the work of each member of the team. • Written work recognises the contributions of others.	• Some ideas were presented but were not fully explained. • Ideas from others were occasionally considered. • Poor teamwork held back the progress of each member of the team. • Written work does not recognise others' contributions.

Explanation of the success criteria

Use this table to make notes on how do well during the course.

Criteria	Explanation
Experiments were conducted in a structured manner.	• Experiments are carried out in a well ordered approach such that further modifications can be made, for example starting with the longest bristles and shaping/chopping them such that no modification hinders the next. • Detailed outcomes are noted for each modification, these help to inform further decisions. • The experiments performed help to indicate what makes an effective BristleBot.
Ideas for further improving the experiments and BristleBots are fully discussed.	
Work produced could be easily recreated by another to achieve the same results.	
Others' ideas were respected and considered.	

Glossary of Keywords

Word	Definition
Battery	An energy storage device that stores chemical energy and is able to release it as electrical energy to drive current around an attached circuit. A rechargeable battery uses a reversible chemical reaction to store energy and may be recharged by putting current into the battery rather than taking it out.
Current	Current is a measure of the flow rate of carriers (electrons) in an electrical circuit. It has SI units of Coulombs per second (C/s or $C.s^{-1}$), where the Coulomb is a unit of charge.
Friction	The resistance that one surface or object encounters when moving over another.
LED	Light Emitting Diode. A semiconductor device that when current flows through it converts the electrical energy into light. LED are available in different colours e.g. infrared, red, green, blue and even white. White LEDs are not actually white, they either use a blue LED which is covered in phosphor (Phosphor emits white light when pumped with blue light), or they use a red, green and blue LED.
Microprocessor	A single integrated circuit that contains all the functions of a 'small' computer. These devices are used in many places from washing machines to controlling the radio chip in you smart phone. They act as the brain of many systems.
Motor	A machine, especially one powered by electricity, that supplies motive power for a vehicle or for another device with moving parts.
Proximity sensor	A device that is able to sense the distance between objects. In the BristleBot we use an InfraRed (IR) based proximity sensor. It works by sending out an IR signal from an IR LED (just like a TV remote) and then using an infrared receiver to measure the signal that is reflected from the object. When the object is closer to the receiver more light will be collected by the receiver than when it is further away.
Resistance	Resistance is an electrical quantity that measures how the device or material reduces the electric current flow through it. The resistance is measured in units of Ohms (Ω).
Resistor	An electronic component that resists the flow of electrical current.
Transistor	A tiny electronically controlled switch or amplifier. Billions of these are used to make the integrated circuits inside virtually all modern electronics, from the computers on your desks and in smart phones to the large transistors used to switch on motors in industrial systems.
Velocity	The speed of something in a given direction. The SI units of velocity are (m/s or $m.s^{-1}$)
Vibration motor (Vibromotor)	An improperly balanced motor that vibrates or wobbles.
Voltage	Voltage is a measured in Volts (V) of the energy in Joules (J) carried by the charges in a circuit. It has SI units of Joules per Coulomb (J/C or $J.C^{-1}$).

Tutorial 1 – What is an Engineer? What is a BristleBot?

Objectives

The goals of today's tutorial are:

- To understand the role of engineers in designing new products
- To consider how engineers collaborate in design groups to create the best products
- To design and construct a simple BristleBot
- Consider the purpose and function of the components that make up a BristleBot

What is an engineer?

Engineer?

Points to consider:

- How many different types of engineer can you name?
- What products might engineers have helped to design?
- What would can engineer do in their daily work?
- Can you draw what you imagine an engineer would look like?

Engineering design teams

When engineers are designing and producing a new product they often work in a team, including both different types of engineers but often other specialist workers. Throughout this course and the 'taming the BristleBot' challenge you will be working in small groups to design, modify and develop a BristleBot that can avoid objects and travel in straight line.

Below are descriptions of a number of engineers that often work together when a new product is been designed. Read through this description and consider:

- What skills/qualities do all the engineers have in common?
- How do the key skills, relate to the job description for each engineer?
- Why do you think these engineers need to work together?
- Which engineer do you think your skills most closely align to?

Design Engineer

Job Description:

- Usually works in a team to takes a concept or product specification and turns it into a design that can be built
- Carry out design calculations and modelling
- Produce detailed design specification, drawings and plans
- Make sure everyone in the team is aware of the health and safety risks
- Support hardware and testing engineers in the development and testing of the prototype through to final product

Key Skills:

- Excellent listening, written and oral communication skills in order to facilitate team working and to communicate with the customer
- Mathematics
- Drawing (Artistic)
- Responsible and reliable
- Excellent knowledge of laboratory safety
- Works well in a team

Hardware Engineer

Job Description:

- Takes an engineering design and builds it
- Provide input into the design process with the design engineer
- Support the test engineer in the development of test facilities and in the testing of the hardware

Key Skills:

- Excellent listening, written and oral communication skills in order to document the build and facilitate team working
- An ability to build and construct things
- Well organised
- Works well in a team

Test Engineer

Job Description:

- Develop test protocols and test systems
- Carryout tests on the design and hardware
- Report results back to the design and hardware engineers and the customer

Key Skills:

- Excellent listening, written and oral communication skills in order to document the build and facilitate team working
- Excellent attention to detail
- Works well in a team

Why do you think it is important and beneficial for engineers and scientists to work in teams?

What characteristics will make you a successful team of engineers? (Consider reading the mark scheme)

Bullet point guidelines your team will adhere to throughout the different challenges to ensure they are a successful team and make the most effective BristleBot.

*(Remember as part of the final assessment your tutor will be evaluating how effective your BristleBot is at avoiding objects and your 'teamwork' skills **will be assessed here**)*

Constructing a Simple BristleBot

A BristleBot is a small robotic device that moves 'chaotically'. The key component of the simple BristleBots we will make today and the more advanced version we will produce towards the end of the programme is an Oral-B Pulsar electric toothbrush.

Task: Use Figure 1 to identify and label the main parts of your BristleBot, add the purpose of the component into Table 1.

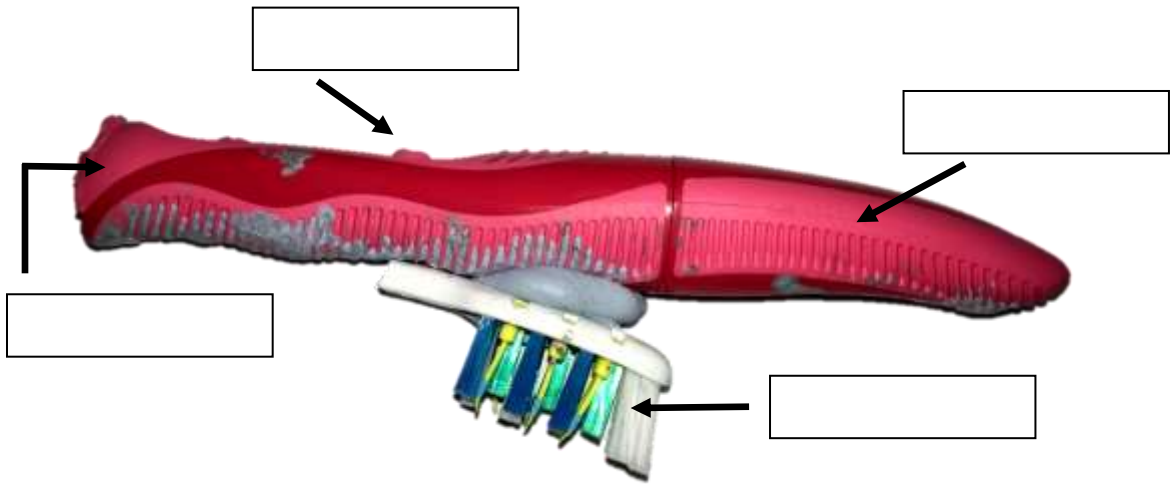


Figure 1 - Simple BristleBot showing functional parts

Parts	Purpose
Switch	
Battery	
Vibration motor	
Toothbrush head	

Table 1 - Simple BristleBot functional parts

Task: In your small engineering design teams, following the guidance given by your tutor you are to make a simple BristleBot device. Once you have made a BristleBot trial turning it on to see the type of movement that it exhibits.

As you are making your simple BristleBot observe its behaviour and make a note of any problems or issues you have during its design and movement.

The Engineering Design Process

Engineers are continually designing new products and modifying old ones to overcome everyday problems and to help us be effective as possible in our everyday lives. This could be from designing a new spaceship to take astronauts to the moon to modifying the screen on a mobile phone to make it easier for people to type messages.

In doing design and development of new products engineers use the Engineering Design Process (EDP) shown in Figure 2. This helps them to structure their thinking and importantly reminds that they have never finished! – Even the best solutions can be improved in the future.

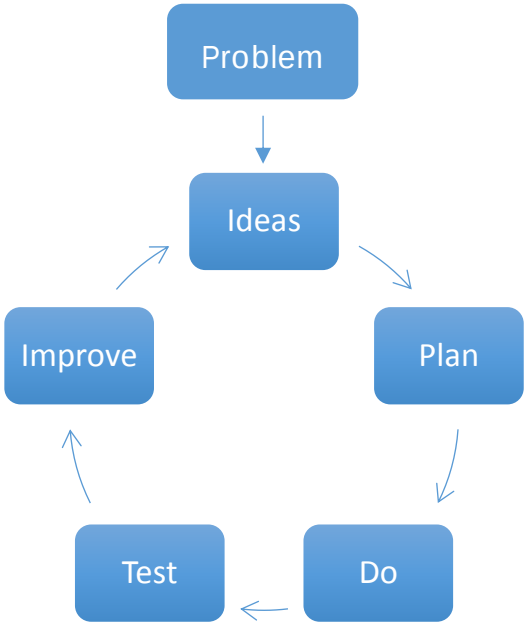


Figure 2 - The engineering design process

Note: Engineers do not always follow the design process in order, for example they might 'Plan' and then take a step back to 'Problem' before continuing with further 'Ideas'.

Task: Write your own descriptions of the keywords in the engineering design process

Problem:

Ideas:

Plan:

Do:

Test:

Improve:

Task: Use the engineering design process to address a number of problems that you identified with your BristleBot. Take each problem in turn and test your improvements and repeat the process as necessary until you are happy with the performance.

Problem:

Ideas:

Plan:

Do:

Test:

Improve:

Second iteration

Problem:

Ideas:

Plan:

Do:

Test:

Improve:

Things to consider:

- The direction of the BristleBot toothbrush head.
- The position and angle at which the toothbrush head is placed.
- Weight distribution, could you add a weight to the BristleBot (blue tac?)

Further Investigation (optional)

You can investigate how the brush head moves the BristleBot forward by using the following steps, this may further help you decide how your simple BristleBot can be improved.

1. Remove the brush head from the Bristle Bot and place it on a flat surface.
1. Depress it slightly with your finger (reasonably gently for a few seconds) then quickly remove your finger and observe what happens.
2. Take note of which direction the brush head moves in. Use this observation to explain how the BristleBot moves.
3. Reassemble your BristleBot and ensure that it is able to stand upright in a resting position and when it moves forward.

What do you observe about the motion of your Simple BristleBot when the vibromotor is turned on? Does it tend to go in a particular direction? Why?

Tip: To observe what is happening when you BristleBot is moving you can use a smart phone camera to video the action. You can also watch the video in slow motion using a free App like Videoshop (www.videoshop.net) which is available for iOS or Android and lets you adjust the play speed.

Task: Use the space below to record your final design

Homework assignment – Tutorial 1

This homework assignment will prepare you for writing the method section of your final assignment.

Task: Write a method of how you produced the final version of your simple BristleBot. This should include any of the modifications that you made and the justifications for these modifications.

Method	This section includes both relevant theory and experimental details. Separate subsections: • Theory and Experiment, may be included. • A brief description of the methodology should be given. This may include a set-up diagram if appropriate. • Don't give lists of instructions. • Don't write chronologically unless that really is the most logical way to present the methodology. • Write in the past tense.	Outline the modifications that you made to the Bristle-Bot. This should give enough detail that someone else could make the same BristleBot.

Top tips:

- Write in full sentences and paragraphs
- Include diagrams/photos as necessary
- Write in the third person (ie. Do not use 'I/we did…')
- Justify the modifications you made using scientific theory as much as possible

Tutorial 2 – Speedy BristleBots

Objectives

The goals of today's tutorial are to:

- Explore how engineers approach problems
- To calculate the speed of a BristleBot
- Improve the speed of the Simple BristleBot

Review of Assignment 1

Throughout the programme, your tutor will provide you with feedback on your homework assignments, this will be helpful in preparing you to get the best possible grade on your final assignment.

Carefully read through the feedback the receive, based on this feedback complete the grid below which will help you to prepare for the next homework assignment and ensure you can improve on each assignment you complete

Two things I did in my first assignment were: 1. 2.
One thing I will aim to improve on my next assignment is:

Improving the Simple BristleBot

You have now designed and developed a simple BristleBot, the challenge is now to make an effective 'speedy' BristleBot. To ensure that your team's BristleBot is going to be effective in this challenge your BristleBot must be able to:

- Travel in a straight line
- Travel at a high speed
- Travel for an extended period of time without falling over

Use the EDP to ensure your BristleBot is as effective as possible.

Testing your BristleBot

One of the characteristics of an effective BristleBot was that is travels at a high speed. We are going to test the BristleBots that we have designed and developed to help analyse what modifications create high speed BristleBots.

To calculate the speed, we will need to measure the distance a BristleBot travels in a particular time. We can do this a number of times with different modifications to help us understand what modifications give the highest speed.

Speed (m/s) = Distance (m) / Time (s)

To ensure the results we record are reliable we should test the same BristleBot a number of times and take an average measurement of the speed recorded.

To calculate the speed, we will need to measure the distance a BristleBot travels in a particular time.

It is easier to measure speed in a straight line. However, unless you are very lucky it is unlikely that your BristleBot will travel in a straight line. You will need to devise a system to ensure that your BristleBot travels in a straight line to simplify speed testing – you can again use the EDP to design the method you are going to use to test the speed of your BristleBot.

BristleBot Modification	Justification for Modification	Distance (m)	Time (s)	Speed (m/s)	Average mean Speed (m/s)
1)					
1)					
1)					
2)					
2)					
2)					
3)					
3)					
3)					

Team _____ best average mean speed: m/s

Team _____ best average mean speed: m/s

The modification which lead to the highest speed BristleBot was:

I think the reason for this was because:

Homework Assignment – Tutorial 2

This homework assignment will prepare you for writing the code that will help to control the movement of your BristleBot in the next tutorial. It will also prepare you for writing the introduction of your final assignment where you will outline the problem you were tasked with and how the engineering design process is a useful structure to designing and developing new products.

Task 1:

Read the passage below about computer coding and answer the corresponding questions.

Programming basics

Adapted from: http://www.programmingbasics.org/en/

Computers aren't very smart. Sure, they can do a lot of math or help you search the Internet. But, if you asked a computer to vacuum the house for you, could it do it? If you asked a computer to draw a picture of a bird for you, would it? A computer would have no idea about what you're saying.

Computers are bad at understanding things. If you don't give them exact instructions, they become confused and make mistakes. Telling a computer what you want it to do is sometimes hard because you have to explain things very carefully.

Because computers don't understand English, you have to give them instructions in special computer languages that computers can understand. There are a number of different languages that computers can understand, these are called codes.

Don't be frustrated if you write some code that a computer doesn't understand. Remember that computers are easily confused, and all the coding instructions have to be exactly right. Computers are very picky! Little mistakes or problems in the code are called bugs and they will result in errors when you run your program.

Most computer codes are an object-oriented languages. That means that the language has objects that have properties and you can give commands to the object. It's just like in real-life. If you have a dog (the object), it's colour (a property) might be black and you can tell (command) your dog to "sit" or to "roll-over." Objects are nice because they allow the programmer to create complex objects out of simpler objects. For example your robotic dog might be constructed out of three objects a head, body and leg objects, each with their own properties and commands.

The computer will follow the first command, and then do the next command, and then the next one, until it reaches the end. This list of instructions is called a "program."

1. Why are computers bad at understanding things?
2. What happens when you give a computer the wrong instructions?
3. What is an object in a computer language?
4. What is a computer programme?

Task 2:

Write an introduction for your final assignment, no more than 500 words.

This should include:

- What is a BristleBot (including the key component parts)
- The problems associated with the simple BristleBot
- The engineering design process and how this is helpful to structure an approach to designing a new product or developing a past product

Introduction	A few paragraphs on the background and motivation to the investigation. Set the scene for the reader and put the work in context. One or two paragraphs on the specific objectives of your investigation; say what you set out to do, what you achieved and why it is important.	Introduce what a BristleBot is and the challenge that you were set to make an effective one. Introduce the engineering design process and how this framed the different design phases of the BristleBot. This may include a diagram of your BristleBot.

Notes

Tutorial 3 – BristleBot Motor Speed

Objectives:

The goals of today's tutorial are:

- To code the LED on the BristleBot to flash
- To activate the motor and control the motor speed with the code
- To remotely control the motors of the BristleBot
- To understand the interaction of 2 motors to enable you to steer the BristleBot

Review of Assignment 2

You will now have received feedback from your tutor on both the 'method' and 'introduction' part of your final assignment.

Carefully, review the two sets of feedback you have received from your tutor and set yourself a target for your next homework assignment.

My target for my next assignment is:

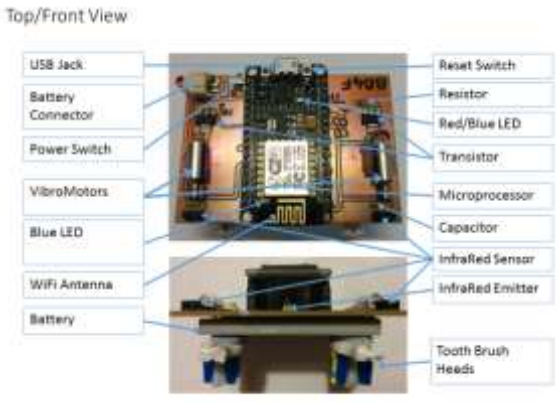
_____

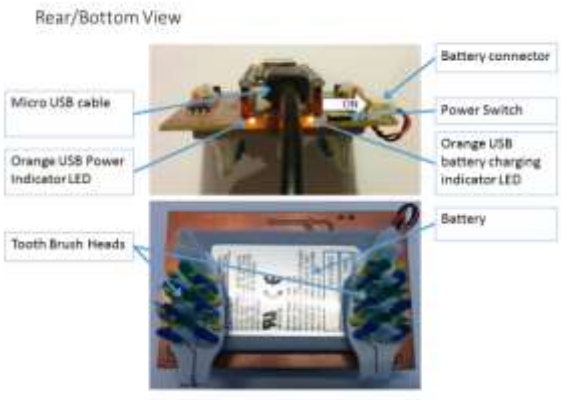
_____

_____

The BristleBot with Microcontroller Brain

As you have noticed the simple BristleBot is pretty good at moving around but does not move in a structured way, to overcome this 'problem' we are going to develop a dual-BristleBot that can be programmed to follow a number of commands.
The dual-BristleBot has many of the same components as the simple BristleBots you developed in tutorial 1 but has additional electronic components that allows you to programme the motion of the toothbrush motors and allows the BristleBot to 'feel' what is around in its environment.





Task: Read the following descriptions in the table below and match them with the component names.

Component	Function
NodeMCU	An computing device incorporating a microcontroller and a WiFi radio.
	A machine that supplies energy for a vehicle
	A component that allows the connection of an external component with the circuit
	A light emitting diode (a semiconductor which glows when a voltage is applied in the correct direction)
	Allows switching between on and off states of the BristleBot
	A component that enables the battery charging functionality
	A component that stores electrical energy
	A device that measures acceleration (such as the Earth's gravitation)
	A device that detects infrared light
	Use to limit the current in the circuit
	An infrared light emitting device
	A certain type of a transistor

Programming a microcontroller

A microcontroller is a small computer, that is used to automatically control products and devices. Each microcontroller each holds a computer programme (code) that controls that the functions that they are take. Microcontrollers are widely used in everyday items such as washing machines, remote controls, microwave ovens, mobile phones and vending machines. A modern car can contain around 40 of them.

Microcontrollers are typically used to support devices that when turned on are constantly doing the same function, whether that is keeping your fridge at the optimum temperature, or maintaining a satellite in orbit

Because of this a typical microcontroller program has two key sections, known as the ***setup()*** and ***loop()*** functions as shown in Figure 3. The code inside the ***setup*** function is only executed one when the power is first applied to the microcontroller, this is to 'setup' the operation of the programme. Then code inside the ***loop*** function is repeated over and over again until the power is removed. This is where you would put the code that controls the function that the microcontroller does again and again (for example monitoring of the fridge temperature and deciding whether to turn the cooling motor on or not).

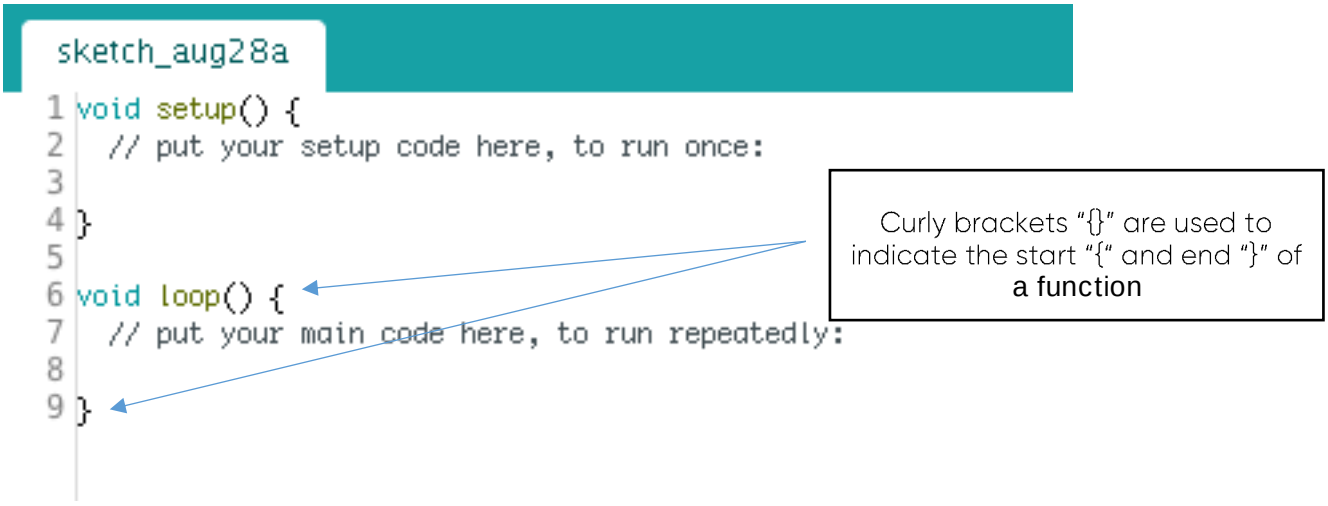


Figure 3 - Arduino microcontroller code structure showing the setup and loop functions.

Throughout the next tutorials we are going to learn how to programme a Microcontroller, this will allow us to 'control' the BristleBot in a way that will mean its motion will no longer be so 'chaotic'.

Flashing LEDs

In computer programming writing a program that prints "hello world" to the screen is the standard way to test that your programming environment is correctly setup and working.
Unfortunately, most microcontroller applications, such as the BristleBot, don't have screens, however they do often have one or more LEDs.

So to test our programme we will turn and LED on and off repeatedly by creating a "blink" program.

1) Follow the steps in Exercise 1 on the support website to create and download the "Blink" program to the BristleBot.

Notes:
(You will need to be able to do a similar process again, to download code)

2) **Once this is working modify the code to do the following, in each case write down what you did to be alter the code.**

- Make the LED flash on and off

- Make the LED flash on and off more quickly

- Make the LED flash on for 5 second, off for 2 seconds and then back on again for 5 seconds.

Task: Upload the **ESP8266_Motor_OnOff** and let the BristleBot run.

What happens to the movement of the BristleBot?

Task: Modify the code by inserting lines into the loop() in the following ways, make note of how you adapt the code.

1) Turn only the right motor on

2) Turn the right motor on for 5 seconds, followed by the left motor on for 5 seconds

Changing motor speed

In the previous tasks you simply turned the motor on at full power. However to control the BristleBot we are going to have to be able to control the power sent to each motor and hence the speed of the motors. With a Microcontroller this is problematic as a digital device can only set the output pins to be either on (full power) or off. However, our microcontroller can turn it is output on and off at a much faster rate than the motors can respond. We can use this property to control the power to the motor, by rapidly turning the output on and off, at a frequency of 400Hz, and varying the ratio of the on time to off time. The motor then only responds to the average of this signal.

The code we are going to use to change the motor speed is called **analogWrite(pin, dutyCycle)** where the parameter **pin** defines the output pin and **dutyCycle** which controls the power ranges from 0 (0%) off to 255 (100%) full power.

To test **analogWrite(pin, dutyCycle)** we are first going to use it to control the brightness of an LED. *(Unlike the motor, the LED is easily able to turn on and off at 400Hz, however it is your eyes that do the averaging in this case as they are only able to detect changes up to around 20Hz).*

Task: To test your eye response modify the '**blink**' code by reducing the on and off delays until you can no longer see that the LED is flashing on and off. Write here, how you modified the code to the point where you were unable to see the LED flashing on and off:

What was the minimum On-Off period (T) that meant you were unable to see the flashing?

Task: Now change the pin to one of the motor pins and observe the behaviour – see if you can alter the code to make the motors turn at:

- Half the speed

- Double the speed

Tip: This exercise requires you to call analogWrite() many times incrementing the dutyCycle parameter by 5 each time. We could do this by writing the analogWrite() function as many times as we required it, as shown below, however this would be rather time consuming.

```
analogWrite(0);
delay(100);
analogWrite(5);
delay(100);
analogWrite(10);
delay(100);
...
analogWrite(255);
```

To simplify the coding of repetitive tasks we use loops. If we know how many times we want to carry out a particular action then we can use a for loop. Which in Arduino has the following syntax

```
for (int i=0; i<N; i++) {
   myfunction(i);
}
```

Where N is the number of times to execute the code inside the for loop. A for loop example increases the count variable from 0 to 100 in steps of 10. It also creates an array containing the numbers [10,20,30,40,50,60,70,80,90,100]. Note the for loop iterator **i** is used to index the **dataArray**. To access a particular element from the array e.g. the 4th element where 40 is stored we use **element = dataArray[3]**. Note in **c** we start counting from 0 so the 4th element is at index 3.

```
int count = 0;
int dataArray[10];
for (int i=0; i<10; i++) {
   count = count + 10;
dataArray[i] = count;
}
```

Sometimes we don't know how many times we need to repeat something but would like to exit the loop when a condition is met. For this we use a while loop, which has the following syntax

```
while (condition) {
....myfunction();
}
```

A while loop example that also increases the count from 0 to 100

```
int count = 0;
while (count < 100) {
....count = count + 10;
}
```

It is fine to code the motors in this way to change the speed, however it is not very helpful if you want the BristleBot to change speed throughout the time you are using it and be able to adapt its speed based on its surroundings. So we are now going to see how we can use the WiFi connection to update the system parameters and control the BristleBot remotely.

Using WiFi to control the BristleBot

The microprocessor on the BristleBot has a WiFi radio that allows it to either connect to the internet, this will allow us to control the BristleBot from a web page.

Before connecting to the BristleBot we need to download the code that runs the access point and web server on the BristleBot. Use Arduino to download the **ESP8266_BristleBot** code.

To connect the BristleBot go to the WiFi settings on your device and connect to the access point
SSID: BristleBot_APxx
Password: bristlebotcontrol
Once connected open your web browser and type in the following IP address
http://192.168.4.1

Assuming you successfully connected to the BristleBot it should return a welcome page with several options. For motor control click the link to the **complete control** page

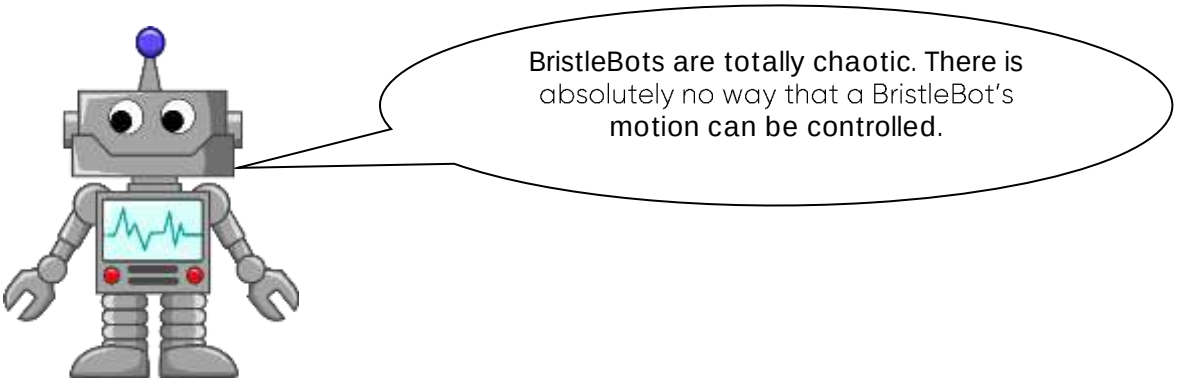
Task: See if you can drive your Bristle bot in a straight line using the left and right motor slider controls.

Explain how you were able to steer the BristleBot using this system, including the sequence you had to enter, does this make you think that one motor is more power than the other, record also any physical changes that you had to make the position of the BristleBot components.

Tip: Ideally we would like both motors to be at similar power levels when travelling in a straight line as this gives us plenty of range on each motor when steering. If the power levels for straight line travel are very different then you might like to consider experimenting with changing the position of the brushes and using Bluetack as a moveable weight to balance the BristleBot in conjunction with the motor power to achieve straight line travel.

Homework Assignment – Tutorial 3

This assignment will focus on the 'communication' skill assessed in the final assignment.



Above is a statement about BristleBots. Your assignment is to write a short essay (around 500 words) explaining whether or not you agree, make sure you give well-reasoned justifications to support.

If you think it's possible to control the movement of a BristleBot, include how you were able to control the movement and any further improvements that you think could be made.

Tutorial 4 – Making BristleBot see

Objectives:

The goals of today's tutorial are:

- To test and calibrate the operation of the proximity sensors
- To use the proximity information to allow the Bristle Bot to avoid obstacles
- To consider what makes a good scientific report

Review of Assignment 3

Carefully review the feedback you have received from your tutor on your assignment. This assignment focused on the communication skill of the final assignment, the marking criteria for this skills can be found below.

Tick which criteria you think you demonstrated in your assignment:

Communication	• A clear set of arguments are made that support and justify the stance that was taken. • The arguments were well evidenced • Language is concise, clear and engaging. • Appropriate technical vocabulary is confidently and correctly used.	• The arguments do not fully support or justify the stance taken • The arguments/statements lacked clear supporting evidence • Language is mostly clear but missing some detail. • Some attempt is made to use technical vocabulary.

Write below one way in which you could have improved your assignment:

Proximity sensors

The proximity sensor on the BristleBot is used to estimate the distance from the BristleBot to objects in front of it. It uses an Infra Red (IR) LED on the BristleBot that emits infrared light and much like a torch illuminates the object in front of the BristleBot. Some of the light is then reflected from the obstacles and detected by the proximity sensors on the underside of the BristleBot.

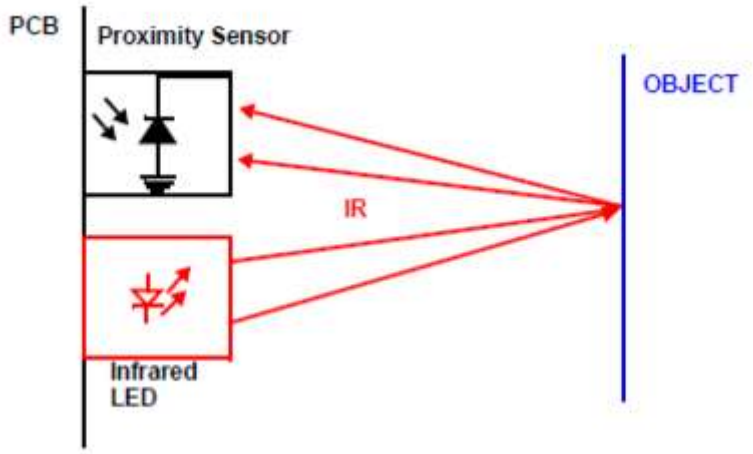


Figure 6 – Diagram showing how a proximity sensor works

These act like the eyes of the BristleBot, however, unlike your eyes which produce images the Proximity sensors only measure the intensity (brightness) of the received light. The intensity readings on the sensors are inversely proportional to the distance between the obstacle and the BristleBot.

As the BristleBot is closer to an object, how much light will be reflected back?

As the BristleBot is further away from an object, how much light will be reflected back?

What else might effect the amount of light that is reflected back?

Task: Follow the instructions in Exercise 2 on the support website using the **ESP8266_Prox_Serial** code and observe how the proximity sensor readings vary with different obstacles placed in front of the proximity sensors of the BristleBot.

Take notes of the results and repeat the experiment with different coloured obstacles.
Do you see varying readings although those obstacles share the same distance from the sensor?

To allow the BristleBot to sense which side the object is closest to it is equipped with two Proximity sensors one on each side (like we have eyes on both sides of our head).
Based on the readings we obtain from left and right proximity sensors, we can make the robot turn appropriately to avoid colliding into obstacles.

Task: To calibrate the sensors in terms of distance place an obstacle in front of the proximity sensors and vary the distance recording the sensor reading. Repeat the experiment three times with different coloured or textured obstacles.

Calibrate - To mark units of measurement on an instrument such so that it can measure accurately

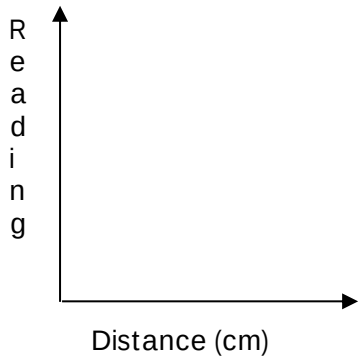
Colour/Texture								
Distance (cm)	Left	Right	Left	Right	Left	Right	Left	Right

Which material do you think gave the best results? Why?

It is more natural for us to use units of distance (cm) when designing when we want the BristleBot to change direction. However it is important to remember that the code in the microprocessor only has the raw sensor readings. Therefore, we are going to have to be able to convert between the reading of distance and the raw reading to give the computer code.

The best way to do this is to plot the results for the best material above, then you can read off what the raw reading will be for each distance.

Task: Plot the results above for your best material.



What would the raw reading be if the BristleBot was the following distances from an object?

- 2 cm: _____
- 5 cm: _____
- 7.5 cm: _____
- 9 cm: _____

Avoiding Obstacles

Now that we have a calibration plot for the left and right Proximity Sensor we can use this to control the BristleBot so that it avoids obstacles.

Task: Save the **ESP8266_Prox_Serial** with a new name and then modify it so that the vibro-motors are turned on. Use the power setting that you determined in the "Making it go Straight:" task so that the BristleBot will travel in a straight line. Add a conditional statement in the loop after the Proximity is read to stop the vibro-motor when the obstacle is within 5cm.

Repeat this experiment several times to test how reliable the obstacle detection is.

Try testing for a range of distances.

Remember that the colour and texture of the obstacle is important.

Note down the performance of your test eg. 90% of the time the BristleBot stopped within ±1cm of the target distance.

Tip: This exercise requires you to use a conditional statement. A conditional statement is a piece of code that first tests whether a condition is true before executing the code inside it. Here we want you check first if the proximity is less than 5cm and if true stop the motors.

The **if** statement, which has the following syntax in **c** provides this behaviour

```
if ([condition]) {
   [do.....];
}
```

For example, if the position variable is less than 10, then stop motor.

```
if (position < 10) {
   analogWrite(pin,0);          // Stop Motor
}
```

Sometimes we would like to do one thing if the condition is true and anther if it is false. The if, then, else statement, which has the following syntax in c allows for this

```
if ([condition]) {
   [do.....];
}
else {
   [do.....];
}
```

For example, if the position variable is less than 10, then stop motor, else start the motor.

```
if (position < 10) {
   analogWrite(pin,0);          // Stop Motor
}
else {
   analogWrite(pin,128);        // Start Motor;
}
```

The condition often makes use of comparison operators e.g.

```
A < B        True if A less than B
```

A <= B True if A less than or equal to B

A > B True if A greater than B

A >= B True if A greater than or equal to B

A == B True if A equal to B

A != B True if A not equal to B

Logical operators are also used

A & B True if A **and** B are both true

A | B True if either A **or** B or both are true

!A Logical Not. If A is true it changes it to false and vice versa

Combinations of comparison and logical operations can also be used. E.g.

(A < 10) & (B < 7) True if A is less than 10 and B is less than 7

Homework Assignment – Tutorial 4

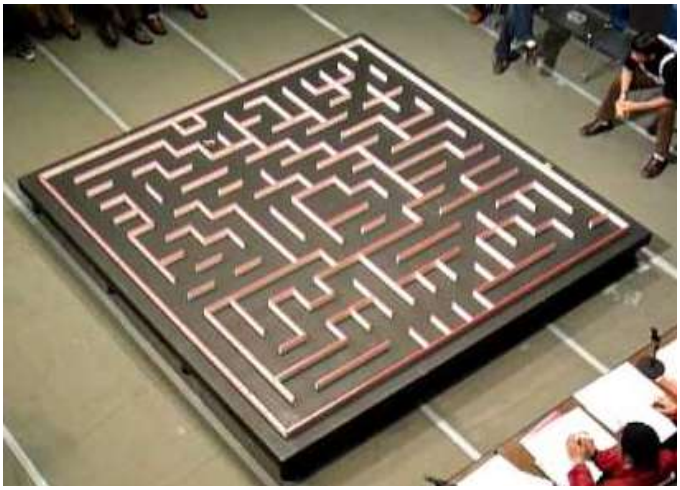
This homework assignment will form a draft for your final assignment. In this assignment you will be writing the first parts of the final assignment.

You should write the 'Introduction', 'Method' and 'Results and discussion' part of the final assignment based on the current BristleBot that you have designed.

Section	Description	What to include	EDP
Introduction	A few paragraphs on the background and motivation to the investigation. Set the scene for the reader and put the work in context. One or two paragraphs on the specific objectives of your investigation; say what you set out to do, what you achieved and why it is important.	Introduce what a BristleBot is and the challenge that you were set to make an effective one. Introduce the engineering design process and how this framed the different design phases of the Bristle-Bot. This may include a diagram of your BristleBot.	Problem
Methods	This section includes both relevant theory and experimental details. Separate subsections, e.g. Theory and Experiment, may be included. A brief description of the methodology should be given. This may include a set-up diagram if appropriate. Don't give lists of instructions. Don't write chronologically unless that really is the most logical way to present the methodology. Write in the past tense.	Outline the modifications that you made to the BristleBot (especially to allow it to travel in a straight line and 'see' objects') This should give enough detail that someone else could make the same BristleBot.	Ideas Plan Do (for each of the modifications)
Results and discussion	**Results:** The main section of the report presenting an overview of what happened. **Discussion:** A brief interpretation of the results which give a reasoned argument to explain your observations. A separate section discussing possibilities for future work may be appropriate.	**Results:** Give a description of the final Bristle-Bot and outline how the Bristle-Bot performed in the final maze challenge. **Discussion:** Give reasons to say why you made the modifications you choose to the Bristle-Bot and explain what effect these had on the Bristle-Bo. Discuss how you would design the Bristle-Bot in future to make it more effective and further modifications that you would suggest.	Improve

Remember to refer back to the feedback you received on your previous homework assignment to ensure you are improving on your previous work.

Tutorial 5 – Taming the BristleBot – Maze Challenge



Objectives:

The goals of today's tutorial are:

- Program the BristleBot so that it goes around an obstacles
- To design the final version of your BristleBot that will avoid obstacles
- To discuss the results of what makes a good BristleBot
- To receive feedback on your draft final assignments and prepare for the final assignment

Review of Assignment 4

Carefully review the feedback you have received from your tutor on your assignment. This assignment formed a draft for your final assignment so it is important you consider this feedback when you write your final assignment.

Below there are a number of things that your tutor will be looking out for in your final assignment, plotted on a 'Brilliant' to 'Improving' scale. Put a cross on the scale to indicate where you think your draft final assignment was for each of these.

Brilliant Things		Things to Improve
I have explained what the EDP is.	←→	I need to explain what the EDP is.
A stranger could recreate my work.	←→	I need to provide more detail.
My spelling and punctuation are excellent.	←→	I have made spelling and punctuation mistakes.
I have explained what a BristleBot is.	←→	I need to clearly explain what a BristleBot is.

List below a number of key things you will need to check when you write your final assignment to ensure you implement the feedback you have received:

E.g. I will need someone to proof read my assignment for spelling and grammar.

You should now have two ways which you steer your BristleBot:

1) Pre-coding the motors to spin at different rates
2) Setting the conditions for the Proximity sensors so the motors change speed to avoid objects

In the final challenge you will need to move your BristleBot around a number of pre-defined objects (a maze). Discuss and investigate with your design group the best way you will be able to do this. Complete the EDP below to show your thinking as your develop your solution.

Problem:	
Ideas:	
Plan:	
Do:	
Improve:	

Automatic Obstacle Avoidance

The challenge here is to program the BristleBot so that it will sense an obstacle, stop before hitting it, turn and move around the object before carrying on.

The BristleBot steers by adjusting the power between the left and right motors. You can use the two motor power sliders on the **Actuators** page to drive the BristleBot directly in this way.

To carry out autonomous operation we would like to abstract the steering control into a set of behaviours, e.g. Swerve, 90 degree turn, U-turn. is controlled by adjusting two parameters the motor power and the steering as well as the duration of the manoeuvre. The actual motor drive values are then calculated from these parameters using the following code. The power (*P*) parameter varies from 0 (motors stopped) to 1 (motors running at the maximum speed determined by the straight line motor calibration values, ***leftMotorMax*** and ***leftMotorMax***) and the steering (*S*) parameter varies from -1 to 1, where negative values indicate a turn towards the left and positive towards the right.

```
if (S < 0) {
        leftmotor = (1+S)*P*leftMotorMax;
        rightmotor = rightMotorMax;
}
else {
        rightmotor = (1-S)*P*rightMotorMax;
        leftmotor = leftMotorMax;
}
```

Task: Select the **Steering** link from the main page. This page lets you enter the calibrated left and right motor speeds for straight line travel, the power (0-100%), the steering (-100% to 100%) to cause the BristleBot to turn and the turn duration (length of time before the motors return the the straight line values).

You will need to carry out a set of experiments in order to investigate the steering and turning abilities of your BristleBot so that you are able to program a set of turning manoeuvres, e.g Swerve, 90 degree turn, U-turn.

Record the required Power, Steering, Duration and achieved turning radius in the table below:

Manoeuvre	Power	Steering	Duration	Turning Radius

Task: You can combine the steering functionality with the proximity sensing to create a program that allows the BristleBot to steer around obstacles.

1) Work out the ideal distance from the object so that the BristleBot has enough space to turn to the side, and change the code accordingly.
2) Add several *if statements* so that the BristleBot will turn to the left when the right proximity sensor reading is greater than the left proximity sensor reading, and vice versa. For the case of equal proximity sensor readings from both sensors you can decide whether the BristleBot should turn to the left or to the right.
3) By adjusting the variables *steer* and *power* you can control how fast the BristleBot will turn when it detects an obstacle.

Experiment with different values and choose the appropriate power and steer value for your preference.

Final Challenge – 'Taming' the BristleBot

You will soon be completing the final challenge of moving your BristleBot around a number of obstacles (a maze) to see how effectively you have been able to 'tame' your BristleBot.

This final challenge will form 25% of your final assessment, to help you achieve the maximum grade on this part of the assignment. Carefully read the adapted mark scheme below which only contains those skills your tutor will be assessing in this challenge.

Make a list of all the things your design team will need to consider, to ensure they are successful in this part of the final assignment:

You tutor will now give you some time to finalise the design of your BristleBot and practice moving your BristleBot around the maze that has been created.

As you work in your design team to finalise your BristleBot design, using the EDP to note down any final modifications you make, starting with the problem that you are trying to overcome. You can use the table below to document these modifications the 'Improve' section of the first modification should feed into 'Problem' section for the next.

	Modification			
	1	2	3	4
Problem				
Ideas				
Plan				
Do				
Improve				

Taming the BristleBot – Results

As part of your final assignment you will need to discuss the results of the 'maze' challenge and discuss how successful your BristleBot was in this challenge. Use the space below to record your observations of what was successful (and not) about our BristleBot. You may also want to note want made the other design team's BristleBot successful so you can consider further improvements that you would make.

Notes

As we discussed before the EDP never ends and you can always further improve the products you have designed. Being as creative as you can note down any further improvements you would like to make to your BristleBot to make it even better. Be ready to sure your best idea with the rest of the group.

Final Assignment – 'Taming the BristleBot'

Your final assignment has two components, the practical element that you complete in tutorial 5 (25%) and also a lab report of how you managed to 'tame' your BristleBot (75%).

Your lab report should be no more than 3 pages in length and should be completed in the lab report template that your tutor will share with you on the VLE.

Your lab report should include details of what a BristleBot is, the EDP, how you 'tamed' the BristleBot and also a discussion of the results of the maze challenge. Further details can be found in the table below which also shows what should be included within the different sections of the lab report.

Section	Description	What to include	Design Process
Abstract	State the objectives and main findings. A summary of what was investigated, how it was investigated, what the outcome was and what the main conclusions were. No more than one paragraph.	Brief summary of the Bristle-Bot challenge, including the modifications you made to the Bristle- Bot. A short overview of how the Bristle-Bot performed in the maze challenge.	
Introduction	A few paragraphs on the background and motivation to the investigation. Set the scene for the reader and put the work in context. One or two paragraphs on the specific objectives of your investigation; say what you set out to do, what you achieved and why it is important.	Introduce what a Bristle-Bot is and the challenge that you were set to make an effective one. Introduce the engineering design process and how this framed the different design phases of the Bristle-Bot. This may include a diagram of your Bristle-Bot.	Problem
Methods	This section includes both relevant theory and experimental details. Separate subsections, e.g. Theory and Experiment, may be included. A brief description of the methodology should be given. This may include a set-up diagram if appropriate. Don't give lists of instructions. Don't write chronologically unless that really is the most logical way to present the methodology. Write in the past tense.	Outline the modifications that you made to the Bristle-Bot. This should give enough detail that someone else could make the same Bristle-Bot.	Ideas Plan Do (for each of the modifications)
Results and discussion	**Results:** The main section of the report presenting an overview of what happened. **Discussion:** A brief interpretation of the results which give a reasoned argument to explain your observations. A separate section discussing possibilities for future work may be appropriate.	**Results:** Give a description of the final Bristle-Bot and outline how the Bristle-Bot performed in the final maze challenge. **Discussion:** Give reasons to say why you made the modifications you choose to the Bristle-Bot and explain what effect these had on the Bristle-Bo. Discuss how you would design the Bristle-Bot in future to make it more effective and further modifications that you would suggest.	Improve
Conclusions	Summary of the main results of the work. (3 – 5 sentences)	Summarise the whole Bristle-Bot challenge.	

What makes a good scientific report?

Before you can write your own scientific report, we need to know what makes a good report.

Note down some points on what you think will make a good scientific report:

Example BristleBot report:

The purpose of this practical assignment was to work out how to configure a Bristle-Bot to be able to travel around any set of obstacles without them being known. The criteria to test whether the objective had been achieved or not was whether the Bristle-Bot could successfully maneuver around a simple maze constructed of cardboard boxes. The Bristle-Bot navigating the maze successfully would indicate a 'pass' and any other test result would indicate a 'fail'. The process

Abstract

Bristle-Bot Maze Analysis

XXXXXX, Ben D.
The Scholars Programme – XXXX – Year 8
4/12/2015 and 2[illegible]/2015

The purpose of this practical assignment was to work out how to configure a Bristle-Bot to be able to travel around any set of obstacles without them being known. The criteria to test whether the objective had been achieved or not was whether the Bristle-Bot could successfully maneuver around a simple maze constructed of cardboard boxes. The Bristle-Bot navigating the maze successfully would indicate a 'pass' and any other test result would indicate a 'fail'. The process to create a suitable Bristle-Bot involved overcoming a number of complications including; finding the right balance, altering the robot to move in a straight line and turning the Bot so that it could move through the maze. Unfortunately due to time restrictions the Bristle-Bot was only able to run the maze challenge once, however when in the maze the Bot managed to speed through needing only a few adjustments, enabling the Bot to turn from one direction to the other proved to be the most complicated element of the exercise.

1. Introduction

The Bristle-Bot is based on a breadboard (a thin plastic board that is used to hold electrical components in a circuit) with several other components such as:

- Jumper wires
- A switch
- Photo resistors
- Transistors
- Potentiometers
- A Battery holder

As seen below:

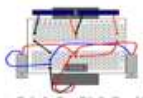


Fig. 1: Bristle-Bot Birds Eye View

Underneath the breadboard were two toothbrush heads that were connected to their own motors, these were the components that enabled the Bristle-Bot to move.

The objective was to create a Bristle-Bot that could navigate its way around a maze without incident. The maze itself was relatively simple in its construction with an approximate distance from start to finish of about a metre and a couple of turns around corners that the Bot needed to negotiate.

Figure 2 shows how the maze was laid out with 'Start' being where the Bot was launched from and the 'Target' being the intended finish line. The Bot could travel either direction as long it was able to move around the obstacles and reach the target.

Fig. 2: Bristle-Bot Maze Overview

Method

Method

A computer was used to configure the speed of the Bristle-Bot. The range of speed went from 0-255 (255 being the fastest) and to begin with the Bot was set to 255.

The breadboard was connected to a computer via USB, this allowed the speed configuration to be uploaded. The Bristle-Bot was able to perform the function of basic movement but this did not incorporate control over the direction in which it travelled and so the Bot did not move in a straight line.

Page 2

In order to navigate the maze the Bot would need to be capable of the following movements:

(1) Move forward 30cm after starting at a 40 degree angle.
(2) Stop after 30cm, turn 80 degrees left by speeding up one motor
(3) Move in a straight line for another 30cm to reach the target

To configure the Bot to achieve this would require setting it to move in a specific direction for a certain amount of time before receiving its next direction. After setting the Bot to move in a straight line by setting both of the motors to 255, it was placed at the beginning of the maze at a 40 degree angle

Figure 3 below shows the path that the Bot took for each of the tests. Different coloured lines have been used to represent each test.

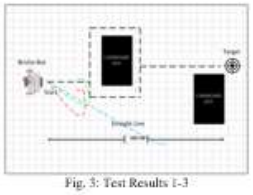


Fig. 3: Test Results 1-3

Given that the direction the Bot travelled in was not consistent across the first three tests a new test was created using masking tape to outline the borders of an area through which the Bot was expected to travel in a straight line.

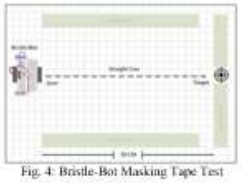


Fig. 4: Bristle-Bot Masking Tape Test

As shown in Figure 4, the Bristle-Bot was placed at the start of the course and the test was to see whether it would travel in a straight line toward the target.

Before test 4-7 were undertaken it was decided that the directional problems could have stemmed from the placement of the toothbrush heads on the underside of the Bristle-Bot. this meant that in order for the robot to travel in a straight line for any distance the toothbrush heads would have to be adjusted.

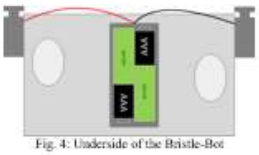


Fig. 4: Underside of the Bristle-Bot

Above is a picture of the underside of the Bristle-Bot with the toothbrush heads at either side of the battery pack. The left hand brush was further up the board than the brush head on the right hand side so it was moved in line.

Below shows the route that the Bot took for the next set of tests.

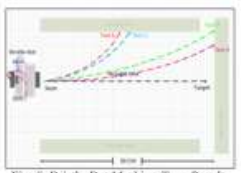


Fig. 5: Bristle-Bot Masking Tape Results

Even after the adjustments that were made the Bot failed to travel towards the target. It seemed that the motor on the right-hand side was more powerful than the motor on the left which meant that they were creating an uneven force. If the Bristle-Bot was to move in a straight line the

In order to navigate the maze the Bot would need to be capable of the following movements:
(1) Move forward 30cm after starting at a 40 degree angle.
(2) Stop after 30cm, turn 80 degrees left by speeding up one motor
(3) Move in a straight line for another 30cm to reach the target

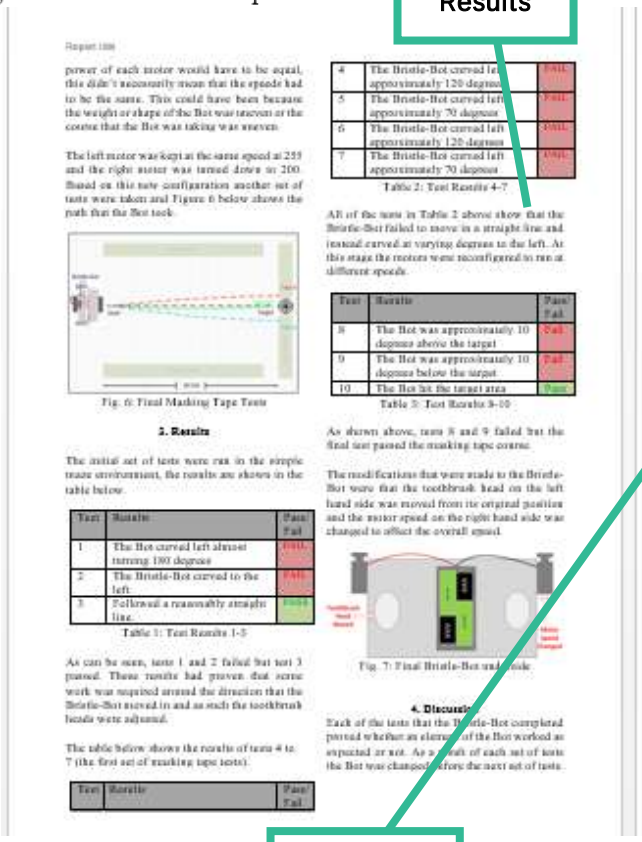
All of the tests in Table 2 above show that the Bristle-Bot failed to move in a straight line and instead curved at varying degrees to the left. At this stage the motors were reconfigured to run at different speeds.

Test	Results	Pass/ Fail
8	The Bot was approximately 10 degrees above the target	Fail
9	The Bot was approximately 10 degrees below the target	Fail
10	The Bot hit the target area	Pass

Table 3: Test Results 8-10

As shown above, tests 8 and 9 failed but the final test passed the masking tape course.

The modifications that were made to the Bristle-Bot were that the toothbrush head on the left hand side was moved from its original position and the motor speed on the right hand side was changed to affect the overall speed.



Results

Discussion

Conclusion

Even after the adjustments that were made the Bot failed to travel towards the target. It seemed that the motor on the right-hand side was more powerful than the motor on the left which meant that they were creating an uneven force. If the

In future experiments the Bristle-Bot would have been more effective if it didn't need programing to command it to complete every action, if it could navigate its way around objects using sensors. The bot would be much more useful if it could sense when something was nearing and then be able to turn itself in a new direction where there was no objects in the way of its path. This would mean that the Bristle-Bot would be much more practical and you could then start it of in any maze and it would be able to find its way through, minus any help.

Using the example above, what makes a good scientific report?

Further tips for your final assignment lab report:

Feedback - Remember to carefully review the feedback you have received throughout the programme on your homework assignments as you have completed a number of assignments that are very similar to the final assignment. Also try to get further feedback as your write your final assignment from your friends, family, teachers and tutor.

Mark scheme – The mark scheme on page X gives you details of the skills that you tutor will be assessing in the final assignment, ensure you read this carefully. You can also self-assess yourself with respect to these skills when you written the first draft of your assignment.

Third Person – Your report should be written in the third person, it should not contain 'I', if necessary it is acceptable to use 'we'.

Figures – If figures are used they should be referenced in the text, and should be numbered in the order they appear in the text. All figures should have a caption but no titles. Figures should be understandable from their caption along.

Plagiarism – In general all material should be original, figures or diagrams from other sources should not be used in the report unless they are essential. If material is used from different cases this should be clearly referenced.

Tutorial 6 – Feedback tutorial

What is the Purpose of Tutorial 6?

- To receive feedback on final assignments.
- To share examples of best practice with the other pupils in your group.
- To write targets for improvement in school lessons.
- To reflect on the programme including what was enjoyed and what was challenging.

Final assignment feedback

What I did well...	What I could have improved on...
•	•
•	•
•	•

My target for future work is...

Reflecting on The Scholars Programme

What did you most enjoy about The Scholars Programme?
•   •   •

What did you find challenging about the programme?	How did you overcome these challenges?
• • •	• • •

Appendix 1 – Referencing correctly

When you get to university, you will need to include references in the assignments that you write, so we would like you to start getting into the habit of referencing in your Brilliant Club assignment. This is really important, because it will help you to avoid plagiarism. Plagiarism is when you take someone else's work or ideas and pass them off as your own. Whether plagiarism is deliberate or accidental, the consequences can be severe. In order to avoid losing marks in your final assignment, or even failing, you must be careful to reference your sources correctly.

What is a reference?

A reference is just a note in your assignment which says if you have referred to or been influenced by another source such as book, website or article. For example, if you use the internet to research a particular subject, and you want to include a specific piece of information from this website, you will need to reference it.

Why should I reference?

Referencing is important in your work for the following reasons:

- It gives credit to the authors of any sources you have referred to or been influenced by.
- It supports the arguments you make in your assignments.
- It demonstrates the variety of sources you have used.
- It helps to prevent you losing marks, or failing, due to plagiarism.

When should I use a reference?

You should use a reference when you:

- Quote directly from another source.
- Summarise or rephrase another piece of work.
- Include a specific statistic or fact from a source.

How do I reference?

There are a number of different ways of referencing, and these often vary depending on what subject you are studying. The most important to thing is to be consistent. This means that you need to stick to the same system throughout your whole assignment. Here is a basic system of referencing that you can use, which consists of the following two parts:

1. **A marker in your assignment**: After you have used a reference in your assignment (you have read something and included it in your work as a quote, or re-written it your own words) you should mark this is in your text with a number, e.g. [1]. The next time you use a reference you should use the next number, e.g. [2].
2. **Bibliography**: This is just a list of the references you have used in your assignment. In the bibliography, you list your references by the numbers you have used, and include as much information as you have about the reference. The list below gives what should be included for different sources.

    a. **Websites** – Author (if possible), title of the web page, website address, [date you accessed it, in square brackets].

    E.g. Dan Snow, 'How did so many soldiers survive the trenches?', http://www.bbc.co.uk/guides/z3kgjxs#zg2dtfr [11 July 2014].

    b. **Books** – Author, date published, title of book (in italics), pages where the information came from.

    E.g. S. Dubner and S. Levitt, (2006) *Freakonomics*, 7-9.

    c. **Articles** – Author, 'title of the article' (with quotation marks), *where the article comes from* (newspaper, journal etc.), date of the article.

    E.g. Maev Kennedy, 'The lights to go out across the UK to mark First World War's centenary', Guardian, 10 July 2014.

Appendix 2 – Using the VLE

VLE username	
VLE password	

Please remember the following key details…

- You are able log into the VLE either through the link on our website (**www.thebrilliantclub.org**) or going directly to the VLE site at (https://portal.thebrilliantclub.org/sign-in).
- Please update your profile with your full name and email address- this will allow you to retrieve forgotten passwords or usernames
- If you forget your log-in details you can request them to be emailed to you by clicking the link on the VLE home page. (If you are still having problems you can email: **schools@thebrilliantclub.org**)

What is the VLE?

The VLE is a virtual learning environment for all pupils on the Scholars Programme it is used for:

- messaging your tutor
- submitting homework
- submitting your final assignment
- accessing resources for your tutorials
- finding out more information about university and careers

How should I use the VLE?

The VLE is a professional academic environment in which pupils are able to message their PhD Tutor. Here are a few things to consider:

- Ensure you keep a professional tone in the messages you send to your tutors.
- Ensure you always reply to your tutors in a timely manner.
- Thank your tutor for the effort they are putting in to give you your feedback etc.
- Submit all homework to your tutor on time.

IMPORTANT: Final assignment

- When you submit your final assignment, please remember that you need to do so through the 'My Activities' tab and not as an attachment to a message.

Notes

Notes

Notes


THE SCHOLARS PROGRAMME
S
THE SCHOLARS PROGRAMME