Make a BBC micro:bit Rover that can be controlled wirelessly using a second micro:bit

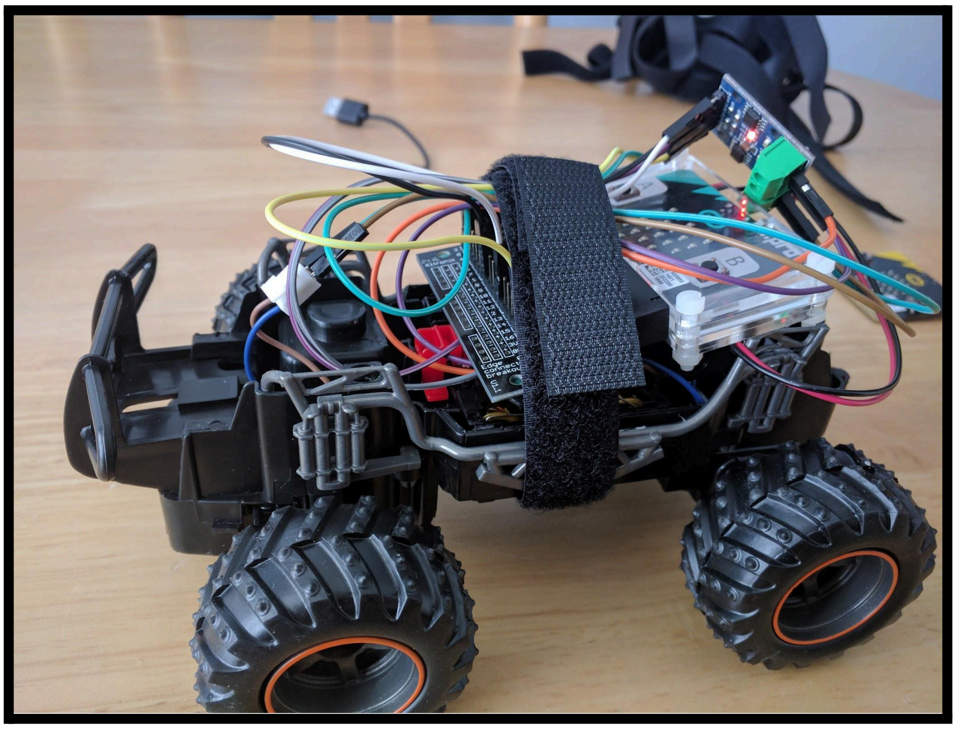
By John Hegarty and Irene Stone
@jhegarty and @istone_irene

Available online at
https://tinyurl.com/microbit-rover

This document is adapted from
"Make a Raspberry Pi Rover that can be controlled over WiFi from anywhere"

By Danny Murray and Stephen Grimes

Available online at
tinyurl.com/roverallover



BBC micro:bit Rover

Contents

Introduction

This document is **aimed at teachers** to help them design a set of lessons for students studying the Junior Cycle Short Course in Coding .

Before the learning experience, students will have experienced how to...

- write pseudocode
- program a micro:bit through creating other projects using the JavaScript blocks editor on the BBC micro:bit website
- use the radio feature on the micro:bits.

Note: This project could be modified to suit students studying the Leaving Certificate Computer Science subject in 2018. The micro:bits can also be programmed using the text version of JavaScript or using Python programming language which would meet some of the draft specifications of the proposed syllabus.

Learning objectives

Students will take a radio controlled toy car and modify it such that they can control its movements through a handheld BBC Micro:bit.

Unit Title: Algorithms and basic programming
Coding may be linked to central features of learning and teaching in Junior Cycle.
Examples of related learning in the course:

> **SOL 17:** The student devises and evaluates strategies for investigating and solving problems using mathematical knowledge, reasoning and skills. Problem solving and computational thinking are central to this course. Students use their mathematical knowledge, skills and understanding when figuring out, evaluating and implementing solutions to particular problems.
>
> **SOL 16:** The student describes, illustrates, interprets, predicts and explains patterns and relationships. Students interpret and describe patterns and relationships as they solve problems and create projects using algorithms and programming languages.
>
> **SOL 20:** The student uses appropriate technologies in meeting a design challenge. Students, either individually or as part of a team, research and discuss the most appropriate technologies to use, to solve problems and deliver solutions.
>
> **SOL 23:** The student brings an idea from conception to realisation.
> Students engage in brainstorming and planning activities, move on to the design, development and test phases, culminating in the creation of a project solution to a particular problem.

Learning outcomes (Junior Cycle Coding specification)

At the end of this unit, students will…

1.4 develop appropriate algorithms using pseudocode

1.5 write code to implement algorithms

1.7 test the code

Assessment

Students will demonstrate their learning when they…

- develop appropriate algorithms using pseudocode; using pen and paper they can instruct each other to move like the car. They will then write code using BBC micro-bit blocks.
- demonstrate to the class their moving rover and describe how they built it and how they programmed it.
- test the code; they will test their pseudo-code by working in pairs. They will test their micro-bit code when they build the car and try controlling its movement with microbit.

Learning Experiences

- Students will modify a radio controlled car by re-wiring it to connect to a micro:bit.
- Students will work in pairs to write pseudocode for “moving like a car”.
- They will write code using micro:bit blocks to control movement of car.

Ideas for student worksheets/ Differentiated learning:

There is lots of scope to cater for different learning needs of students. Examples:

- Some students may require more steps than others for example. Teachers can adapt the amount of steps (as set out in this document) that they will present to students.
- Students could try and figure out the order of the steps.
- Different motor drivers can be used with different levels of difficulty (these are outlined below)
- The micro:bits can be coded with blocks, javascript or python.

Key Skills of Junior Cycle

Key Skill	**Elements**	**How?**
Being Creative	Exploring options and alternatives Stimulating creativity using digital technology	Experimenting with code. Looking for different ways to control the car (e.g. with tilt/ pressing button A etc.) Extension - looking at sequential or parallel blocks?
Working with Others	Co-operating Learning with others Working with others through digital technology	Students can working in pairs or in groups of 3.
Being Literate	Expressing ideas clearly and accurately Developing my spoken language	Students will present to their peers, describing how their code works and what they did.
Being Numerate	Estimating, predicting and calculating Developing a positive disposition towards investigating, reasoning and problem-solving	When writing pseudocode, students could estimate how many seconds it will take for their car to move etc.

Theoretical Framework

Body Syntonic Reasoning

In approaching the challenge of completing this project, students will need to think computationally about a number of abstract ideas in order to effect change in the physical movements of the car.

With the car itself, they will experience the movement that the two motors allow separately and what happens when the movements are combined. One motor allows for forward and backward movement. The second motor turns the wheels without moving the car. Combining these two characteristics allows the car to both move forwards and backwards and turn.

In using the accelerometer on the second microbit, they will use abstract ideas of geometry to give instructions to the microbit attached to the car to effect change in how the car behaves in the physical world.

This learning or thinking based in experience, ties into the model of learning developed by Papert where students used a turtle robot controlled by the Logo programming language to draw pictures. Using their own experience and understanding of movement in the real world they could instruct the turtle to move leaving a pen trail behind. Papert (Papert, 1980) called this Body Syntonic Reasoning. Already having a knowledge and understanding of physical movement both in moving themselves and knowing and understanding the movement in cars gives the students a viewpoint from which they can think about programming.

In this project, students can imagine themselves as a car, reasoning and predicting how the car will behave when given a set of logical, sequenced instructions. Having the physical body syntonic platform to build on they can engage with abstract ideas and experience abstract concepts in a concrete way they find engaging and playful in the challenging manner Papert described as "Hard Fun".

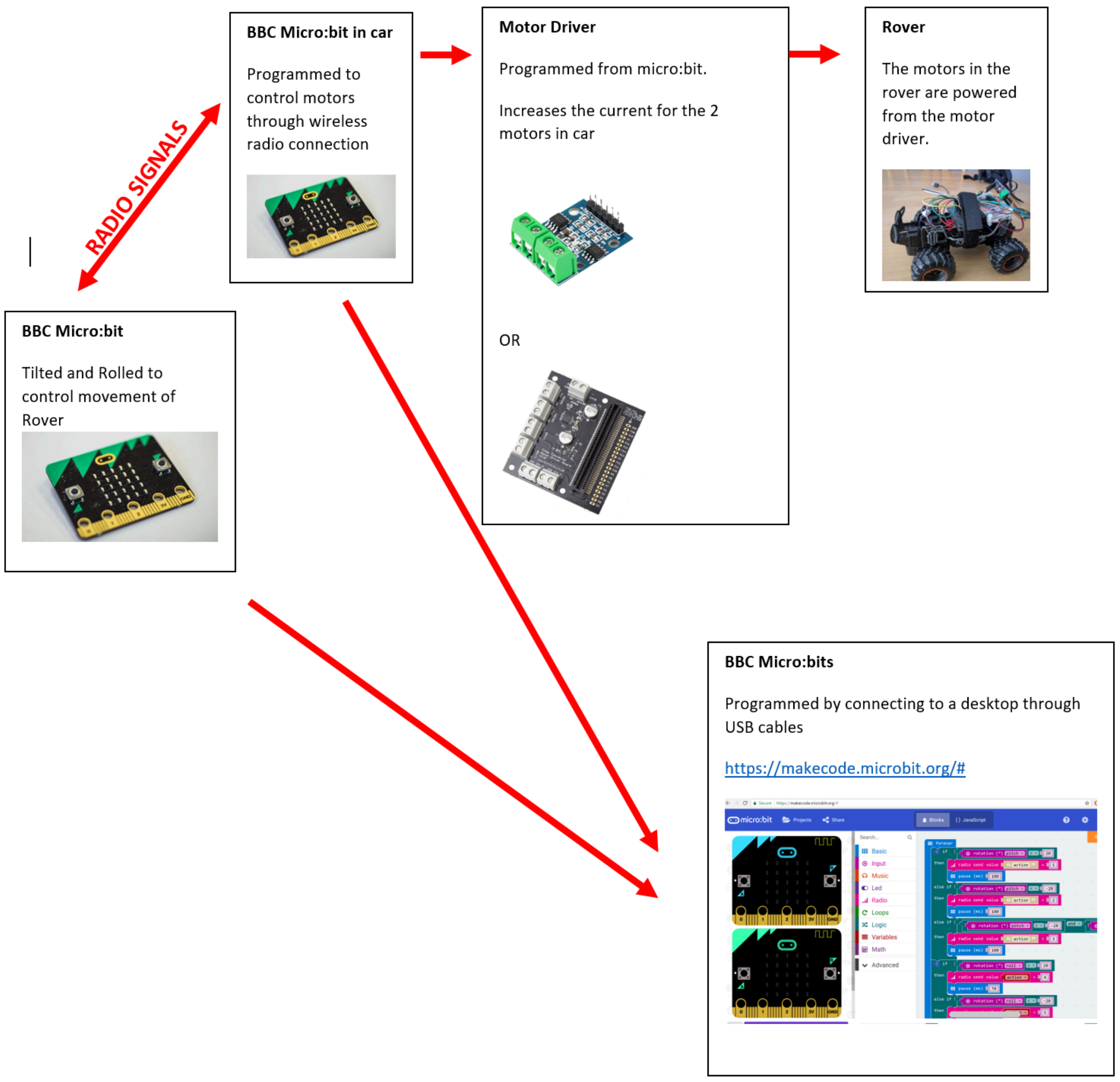
Whereas Papert had his turtle as a constructed computational "object to think with" we will have our micro:bit rover. It will be the vehicle (bad pun intended) we will use to help children learn programming.

Constructionist/ Constructivist Approach

Papert, a constructionist, believed that "knowledge is experience that is acquired through interaction with the world, people and things." (Ackermann, 2001, p.3-4). Students learn through discovery and are active through making a tangible object, e.g. the rover car. The teacher acts as guide or facilitator rather than giving step by step instructions. We recommend that the steps in this guide should not be given directly to students, rather the teacher can scaffold students' learning through giving them starter points for example. Some students could be given more steps than others. From a constructivist view, learners learn most when given tasks that are within their range of learning. This Zone of Proximal Development (ZPD) is the difference between learners trying to figure out the problem alone as opposed to under the guidance of a teacher or more capable peers (Vygotsky, 1978, p. 86).

Ackermann, E. (2001). Piaget's Constructivism, Papert's Constructionism: What's the difference? Retrieved from
http://learning.media.mit.edu/content/publications/EA.Piaget%20_%20Papert.pdf
Papert, S (1980). Mindstorms: Children, Computers, And Powerful Ideas. New York: Basic Books, Inc. Publishers
Vygotsky, L. S. (1978). Mind in society : the development of higher psychological processes. Cambridge: Harvard University Press.

Overview of Project



BBC Micro:bit in car

Programmed to control motors through wireless radio connection

Motor Driver

Programmed from micro:bit.

Increases the current for the 2 motors in car

Rover

The motors in the rover are powered from the motor driver.

BBC Micro:bit

Tilted and Rolled to control movement of Rover

BBC Micro:bits

Programmed by connecting to a desktop through USB cables

https://makecode.microbit.org/#

Parts/Shopping List:

- 2 X BBC micro:bit
- 1 X USB cable
- 2 X power packs for micro:bits
- 1 X Edge connector
- 1 X Motor controller
- Circa 10-12 Jump wires depending on motor controller
- 1 X Remote controlled car

BBC micro:bit

This project requires two BBC micro:bits. One micro:bit will be attached to the Rover and serves as a microcontroller to control the motors. The other micro:bit will be used to communicate instructions to the micro:bit on board the rover. This is done through the built in radio feature on the micro:bits. The micro:bit uses a built in accelerometer detect movement - pitch and roll and depending on the motion detected issue instructions wirelessly to the micro:bit in the rover.

An individual micro:bit can be purchased for approx €20. This includes battery pack and USB cable. The USB cable is needed to connect the micro:bit to a computer so the user can programme it. The battery pack is needed to power the micro:bit when it is disconnected from the computer.
https://www.maplin.ie/p/bbc-microbit-go-with-usb-cable-batteries-and-battery-holder-a41wb
Class set of 10 - Starter Packs, includes USB cables and power packs. Approx £150 (+postage)
https://www.kitronik.co.uk/5616-bbc-microbit-classroom-pack.html

Edge Connector

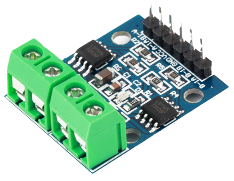
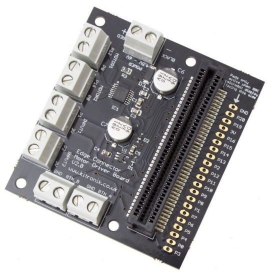
The edge connector gives easy access to all of the pins located on the edge of the BBC micro:bit. This allows you to connect external circuitry to the BBC micro:bit board, e.g. the motor controller needed to drive the motors (and move/steer the car)
Approx €12
https://www.maplin.ie/p/edge-connector-breakout-board-for-bbc-microbit-pre-built-a81uq

Motor Controller

A Motor Controller is a device that acts as intermediary between the microbit and the motors. The micro:bit outputs 3v which is not enough as the motors in the car need 4.5v to run. We can attach the batteries in the car to the motor controller to provide power to the motors. A motor driver is a little current amplifier; the function of motor driver is to take a low-current control signal from the micro:bit and then turn it into a higher-current signal that can drive a motor.

“Let’s say you (the micro:bit) want to lift some heavy stuff but you don't have the muscles. So, you ask for support from the friend (the microcontroller) who spends her mornings and evenings at the gym doing some heavy lifting. However you still have to tell her how to lift the heavy bricks. You are the microcontroller (the BBC micro:bit) and your friend is your motordriver. The bricks is the motor.”

There were two types of motor controller that we recommend be used in this project. One combines with an edge controller and the other has a built in edge controller. Only one is needed per car.

Type of Motor Controller	Pros	Cons	Cost
L9110S 	Very cheap Small motor controller	More steps to get connected	€6 for 10 https://goo.gl/jTfHGx
Used with a separate edge connector with pins	The edge connector can be used in other projects not involving motors	Need a separate edge connector	€6 (+ postage) for edge connector https://goo.gl/1eAz8k
Kitronik Motor Controller & edge connector combo for Microbit 	Easier than other motor controllers to connect - less cables needed No need for separate Edge connector	Expensive Not as flexible to use with other projects not involving motors	Approx €23 https://www.maplin.ie/search/?text=microbit+ OR £13 (plus postage) https://www.kitronik.co.uk/5620-motor-driver-board-for-the-bbc-microbit-v2.html

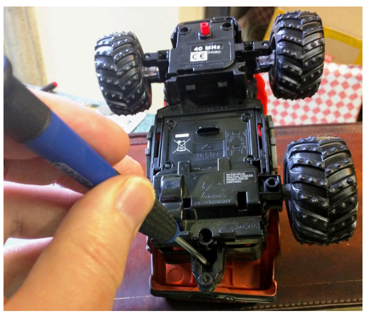
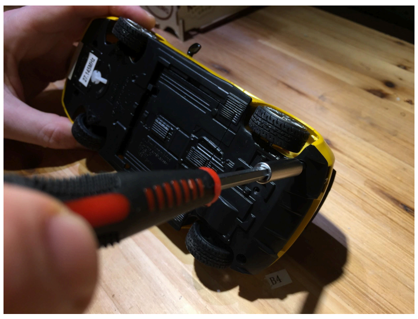
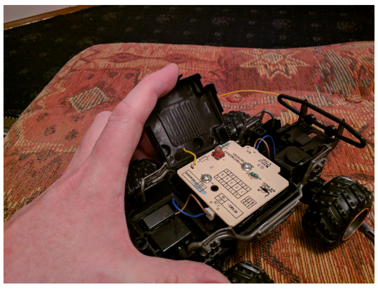
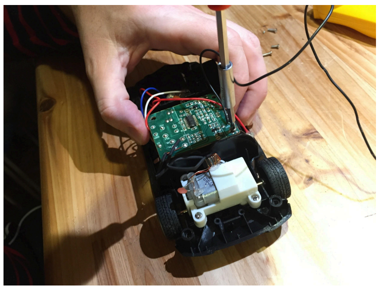
Radio Controlled Car

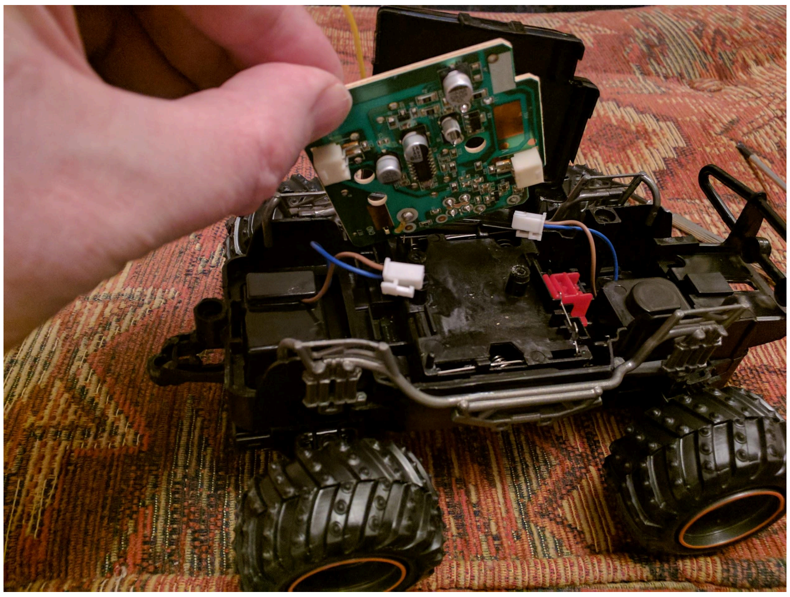
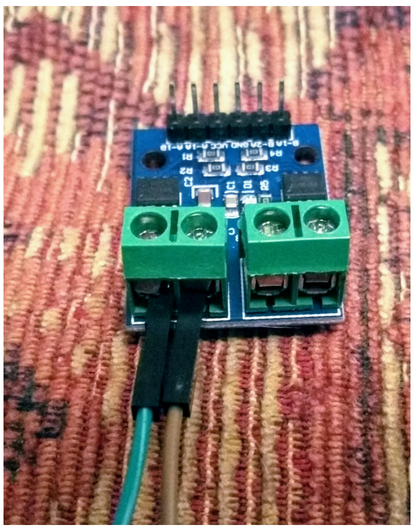
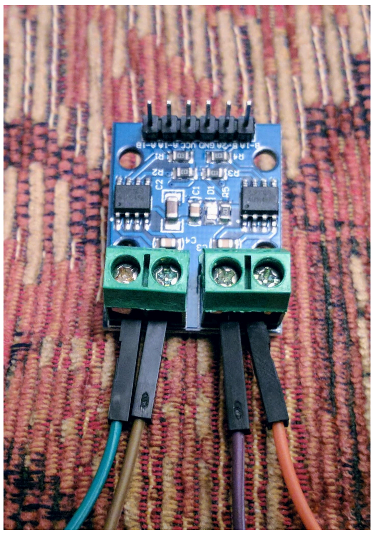
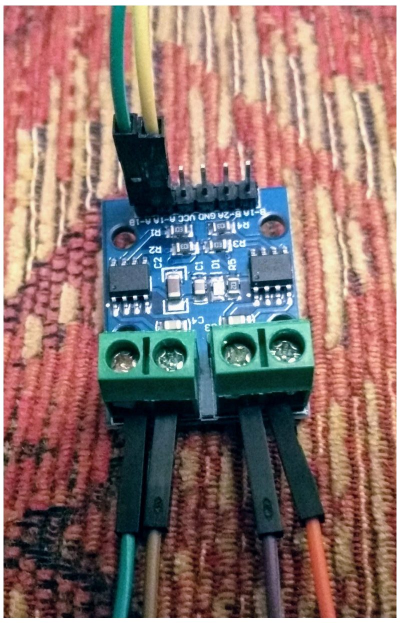
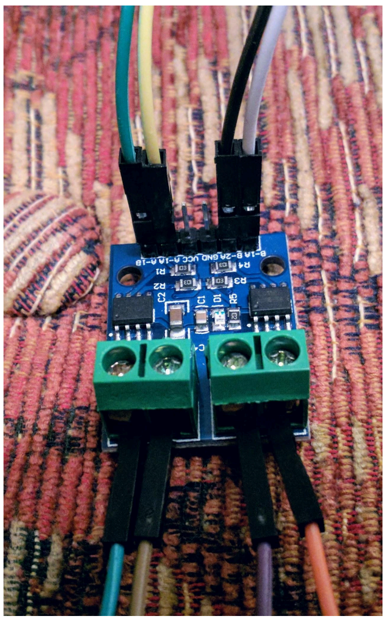
There were two different cars that were used when designing this handout. There are many similar cars that work in much the same way.
E.g. 2 or €20 in Argos, Nov 2017

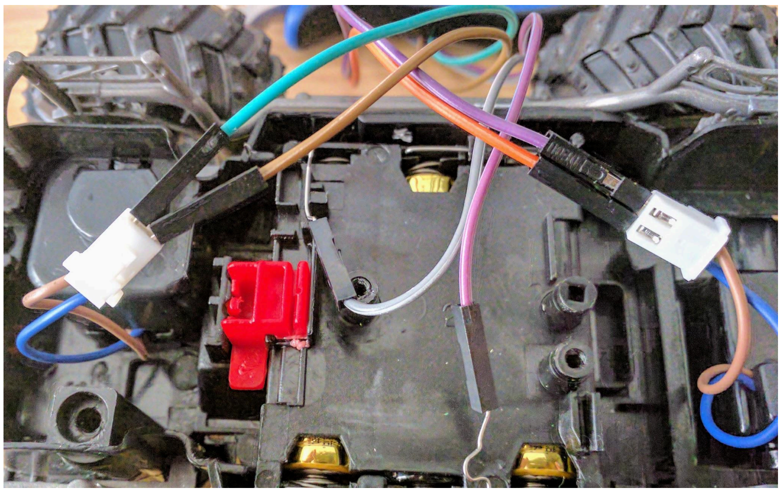
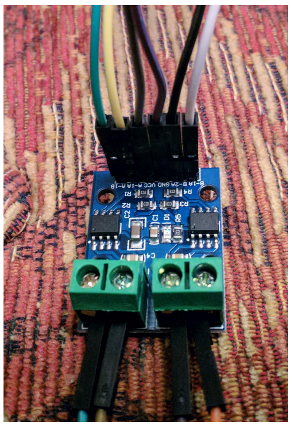
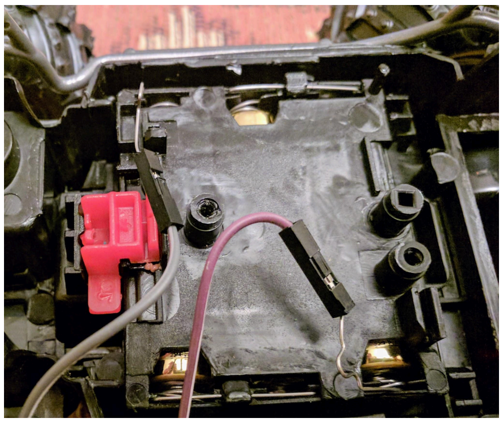
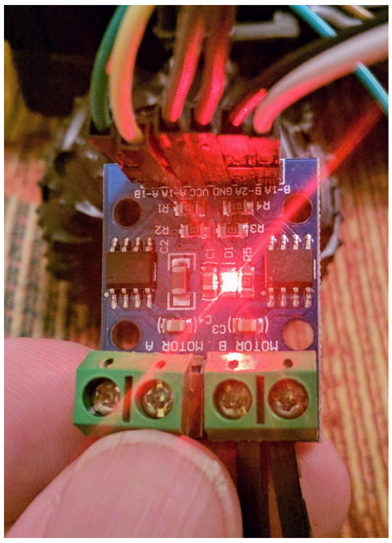
Written instructions with pictures:

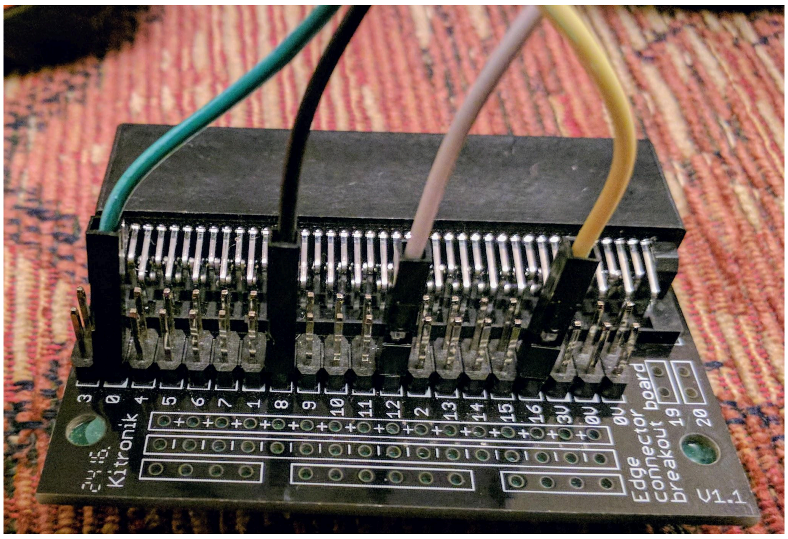
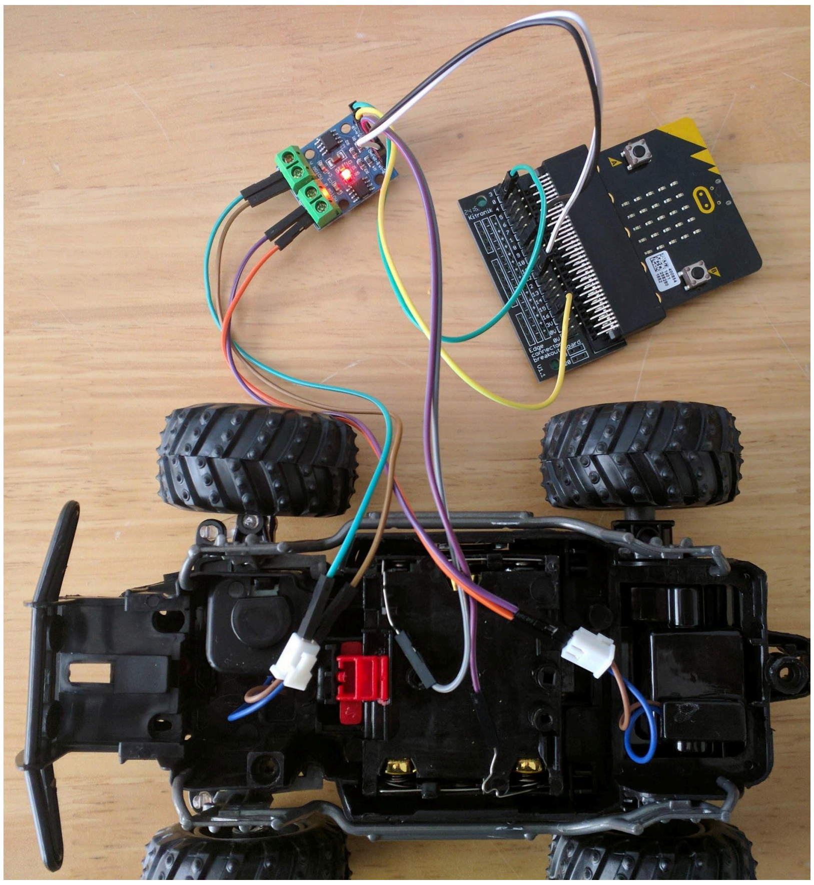
The two different cars are shown below but the steps with both are pretty much the same. The cars you use may have slight variations and there should be enough information here for you to find your way.

Instructions for L9110S motor controller:

Step 1 Remove the top part of the car by undoing the screws underneath holding it in place	 
Step 2 Get access to the circuit board Remove the screws holding it in place	 

Step 3 Lift out the circuit board and ariel Unplug the power cables attaching the motors to the circuit board We will be replacing this circuit board with an alternative controller. Either the cheap L9110S and separate edge controller or the easier to work with and tidier edge connector and motor controller combo from Kitronik - see images above	
If using the Kitronik Motor Controller, then skip to page 13	
Step 4 (For L9110S only) Add the cables to the motor controller to connect to the motors For each motor there will be two cables	 
Step 5 (For L9110S only) Add the cables needed to connect the micro:bit edge connector to the motor controller. Again for each motor there are two cables	 

Step 6 Connect the motor controller to the motors. Make sure the correct pair of cables go to each motor. Don’t worry about the order of the two cables being connected to each motor. If the order is wrong it will still work … but the wheels will turn in the wrong direction. Swapping the cables at that stage will solve the problem. A problem that is easier to solve than prevent.	
Step 7 Connect the car batteries to the motor controller For L9110S connect to the two pins on the controller marked VCC and GND - the middle two of the row of six. A light should come on the controller - if the light doesn’t come on swap the two power cables round.	  

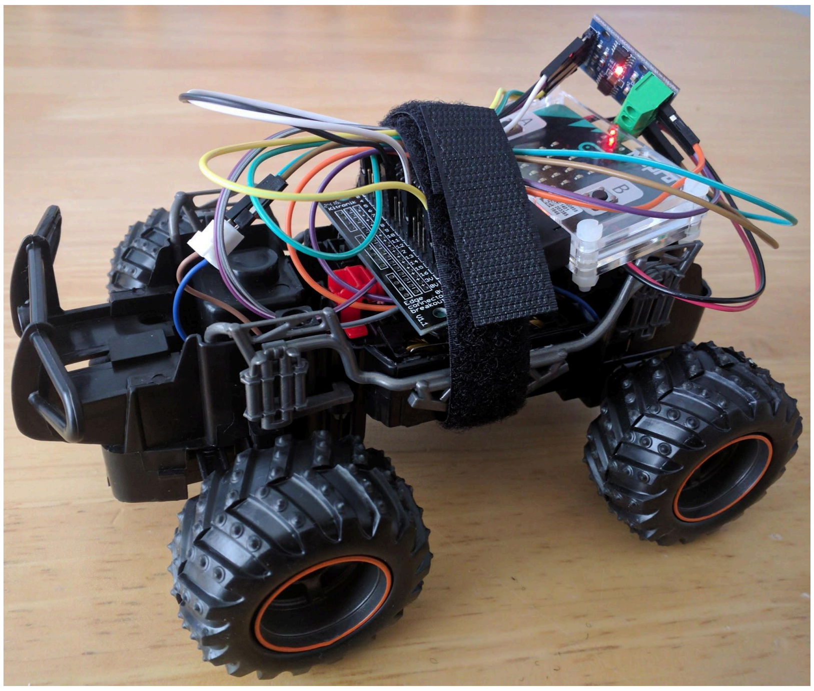
Step 8 (For L9110S only) Connect the motor controller to the Edge connector. Two cables from one side going to pins 8 and 12 (forwards and backwards) and two from the other side going to pins 0 and 16 (left and right). Note the pins aren't in strict numerical order - don't use pin 3 by mistake These cables aren't needed with the Kitronik motor controller & edge connector combo	 3 0 4 5 6 7 1 8 9 10 11 12 2 13 14 15 16 3V 0V 0V Kitronik Edge connector breakout board 19 20 V1.1
Step 9	Program the micro:bits - See next section
Step 10 Connect one microbit to the edge connector with its own battery pack	

Step 11

Use something like a rubber band or strip of velcro to hold the components together.

Be careful to not to have any cables too close to the wheels

Try it out



Instructions for Kitronik motor controller (Steps 1 to 3 same as above)

Step 4

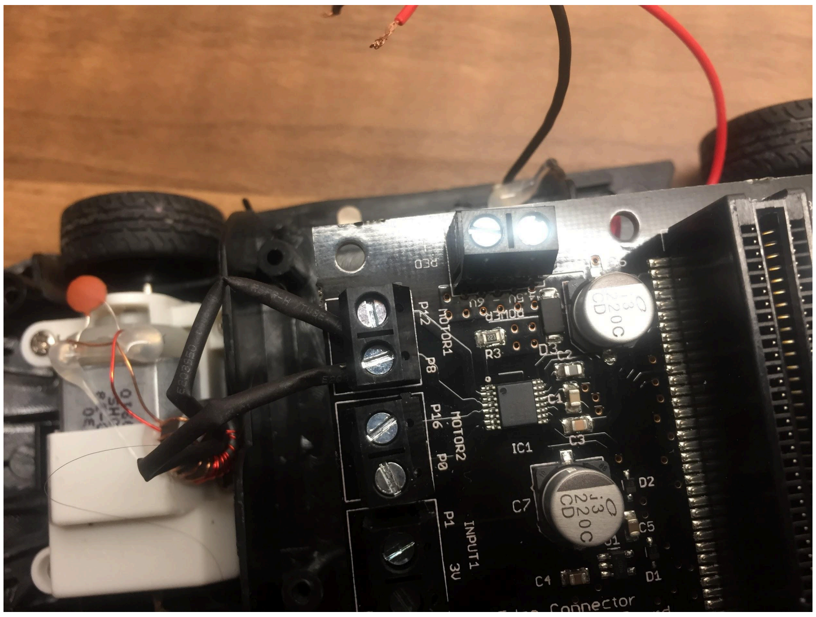
Connect the motor controller to the motors

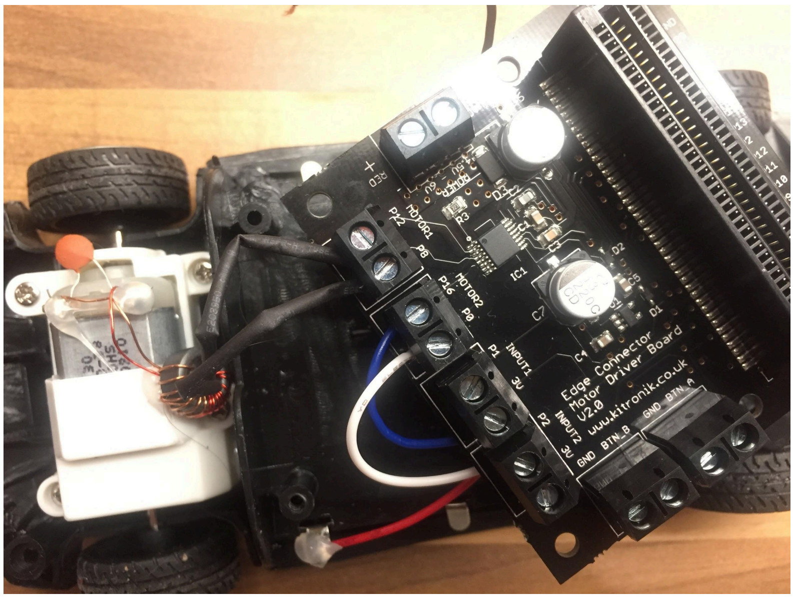
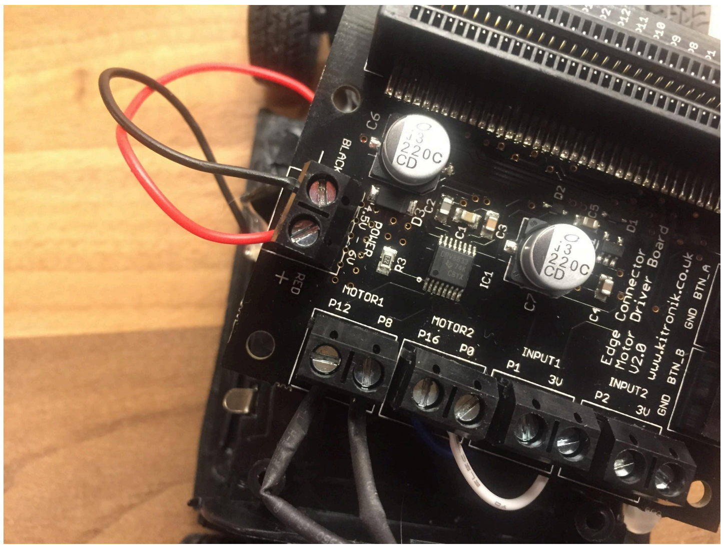
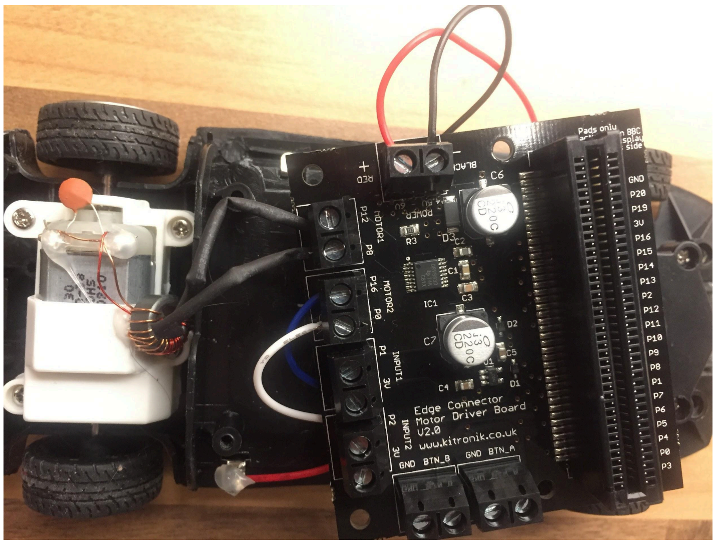
For each motor there will be two cables

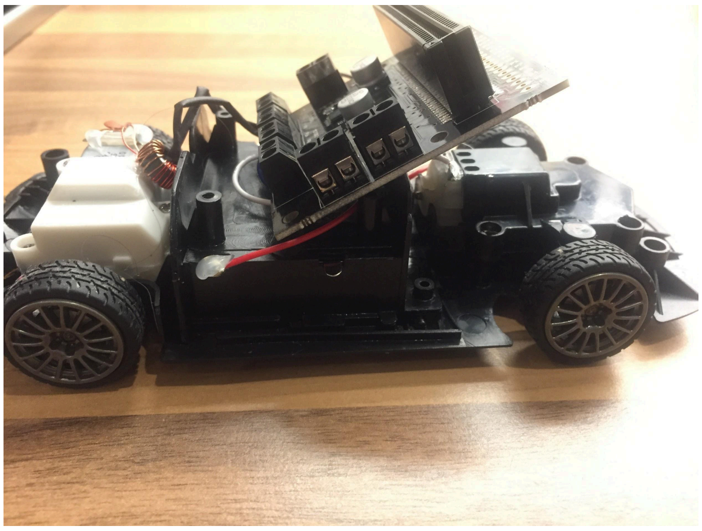
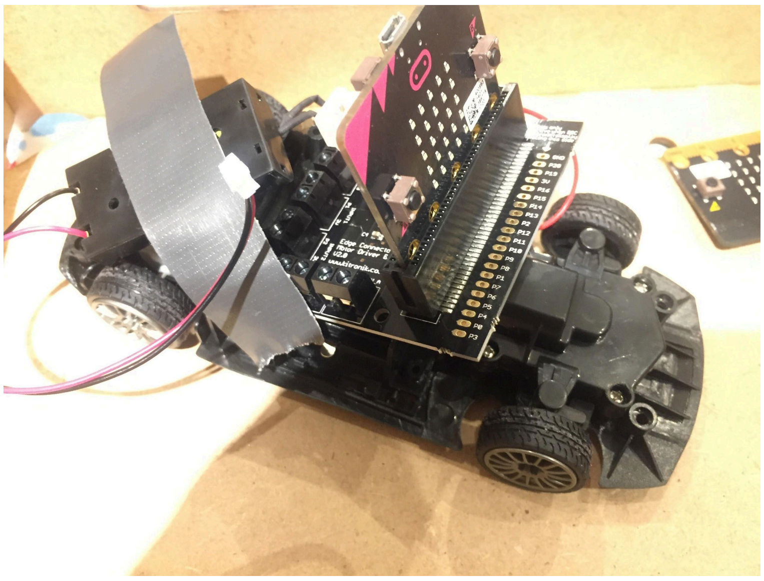
In this picture the back motor is connected to Motor 1 (p12 and p8)

The front motor is connected to Motor 2 (p16 and p0)

Don't worry about the order of the two cables being connected to each motor. If the order is wrong it will still work … but the wheels will turn in the wrong direction. Swapping the cables at that stage will solve the problem. A problem that is easier to solve than prevent.



	
Step 5 Connect the car batteries to the motor controller.	 

	
Step 6	Program the micro:bits - See next section
Step 7 Connect one micro:bit to the Kitronik motor controller edge connector. Note - the micro:bit using the kitronik motor controller/edge connector combo doesn't need its own battery pack. Adding a battery pack might damage the motors as the combination of two battery packs will supply too much power. Use something like a rubber band/ strip of velcro or electric tape to hold the components together. Be careful to not to have any cables too close to the wheels Try it out	

Programming the micro:bits

We will be using two micro:bits

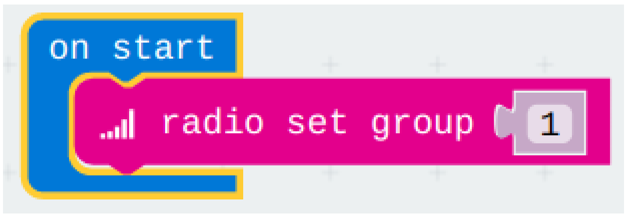
- micro:bit 1 will be used to send messages telling the car to go forward/backward and turn right/left

- micro:bit 2 will be used to send messages to the motor controller which will make the motors move the car as needed.

Sample code available here - https://makecode.microbit.org/_87pcwRHia6Jc

on start - both micro:bits

We will use the built in radio feature of the micro:bit to send instructions from micro:bit 1 and receive instructions on micro:bit 2. Each micro:bit needs to be set to the same group at the start.



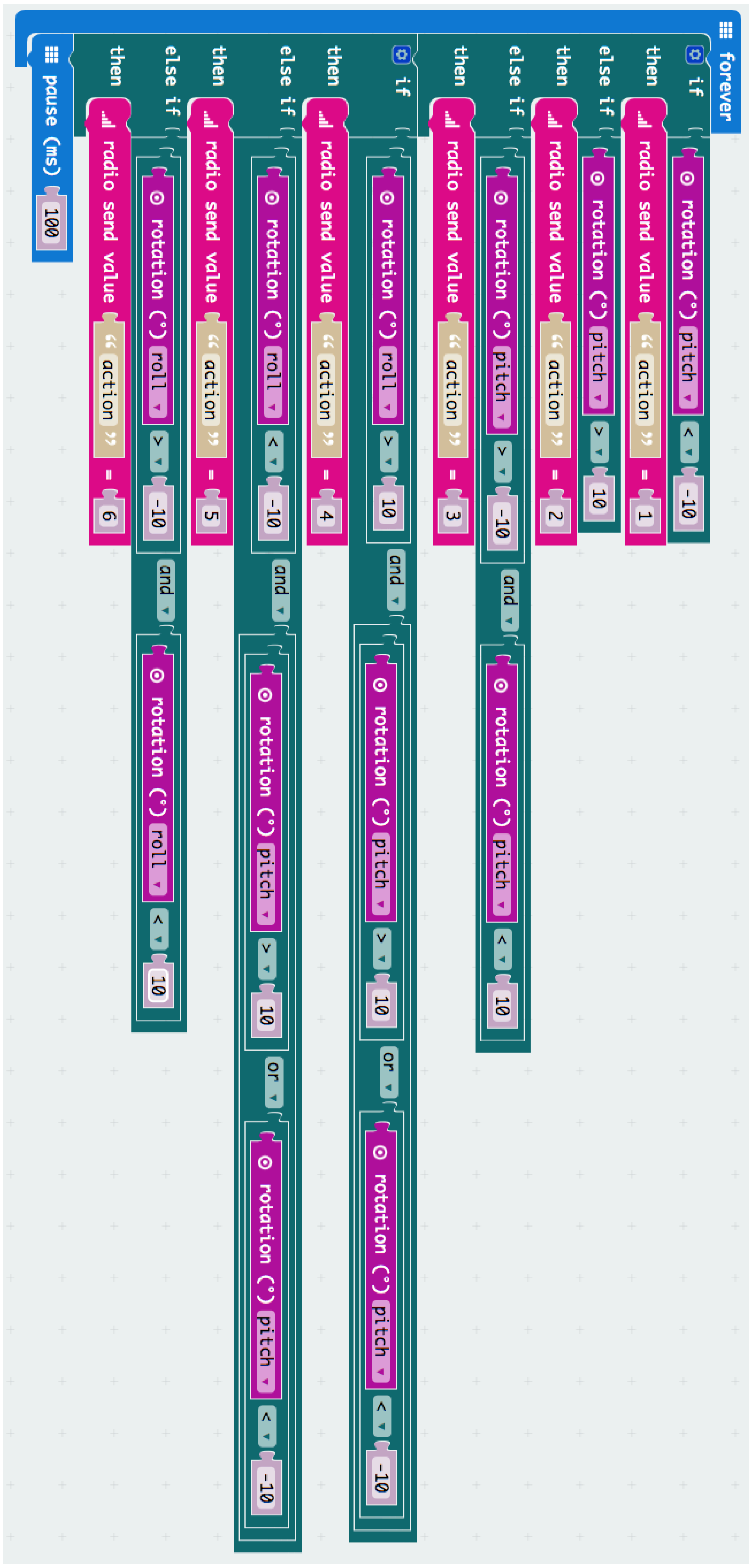
```
on start
    radio set group 1
```

forever on micro:bit 1

We will use the accelerometer built into the micro:bit. The instructions will be

- if pitched forward more than 20 degrees broadcast a 1 (to move forward)
- else if pitched back more than 20 degrees broadcast a 2 (to move backwards)
- else if pitched between -20 and +20 degrees then send a 3 (to stop)

- if rolled (tilted) more than 20 degrees to the right AND pitched forwards or backwards send a 4 (to turn wheels right)
- else if rolled (tilted) more than -20 degrees to the left AND pitched forwards or backwards send a 5 (to turn wheels left)
- else if rolled (tilted) between -20 degrees and +20 degrees (that is not really tilted left or right) send a 6 (to keep the wheels straight)



```
forever
    if rotation (°) pitch < -10
    then
        radio send value "action" = 1
    else if rotation (°) pitch > 10
    then
        radio send value "action" = 2
    else if rotation (°) pitch > -10 and rotation (°) pitch < 10
    then
        radio send value "action" = 3

    if rotation (°) roll > 10 and rotation (°) pitch > 10 or rotation (°) pitch < -10
    then
        radio send value "action" = 4
    else if rotation (°) roll < -10 and rotation (°) pitch > 10 or rotation (°) pitch < -10
    then
        radio send value "action" = 5
    else if rotation (°) roll > -10 and rotation (°) roll < 10
    then
        radio send value "action" = 6
    pause (ms) 100
```

Receiving instructions on micro:bit 2 and

Sending instructions to the motor controller

micro:bit 2 listens for messages broadcast from micro:bit 1 and acts when it receives the message.

Each motor has two connections and is connected through the controller to two pins on the micro:bit

We are going to use the following pins:

- Motor one (forward and backwards) will use pins 8 and 12
- Motor two (left and right) will use pins 0 and 16

Forwards and backwards

When we set pin 8 to the value 1 (on) and 12 to the value 0 (off) the car will move forward.

Switch them round and the motor spins in the opposite direction sending the car backwards - pin 8 to the value 0 and pin 12 to the value 1

Set both pins 8 and 12 to the value 0 and the car stops

Motor 1 Connections Truth Table		
Pin 8	Pin 12	
0	0	Stop
1	0	Spin clockwise (forwards)
0	1	Spin anti clockwise (backwards)
1	1	Stop

Both 0, 0 and 1, 1 will cause the car to stop but we will only use 0, 0

Turning works the same way

When we set pin 0 to the value 1 (on) and 16 to the value 0 (off) the car will turn right.

Switch them round and the motor spins in the opposite direction sending the car to the left - pin 0 to the value 1 and pin 16 to the value 0

Set both pins 0 and 16 to the value 0 and the car wheels stay straight

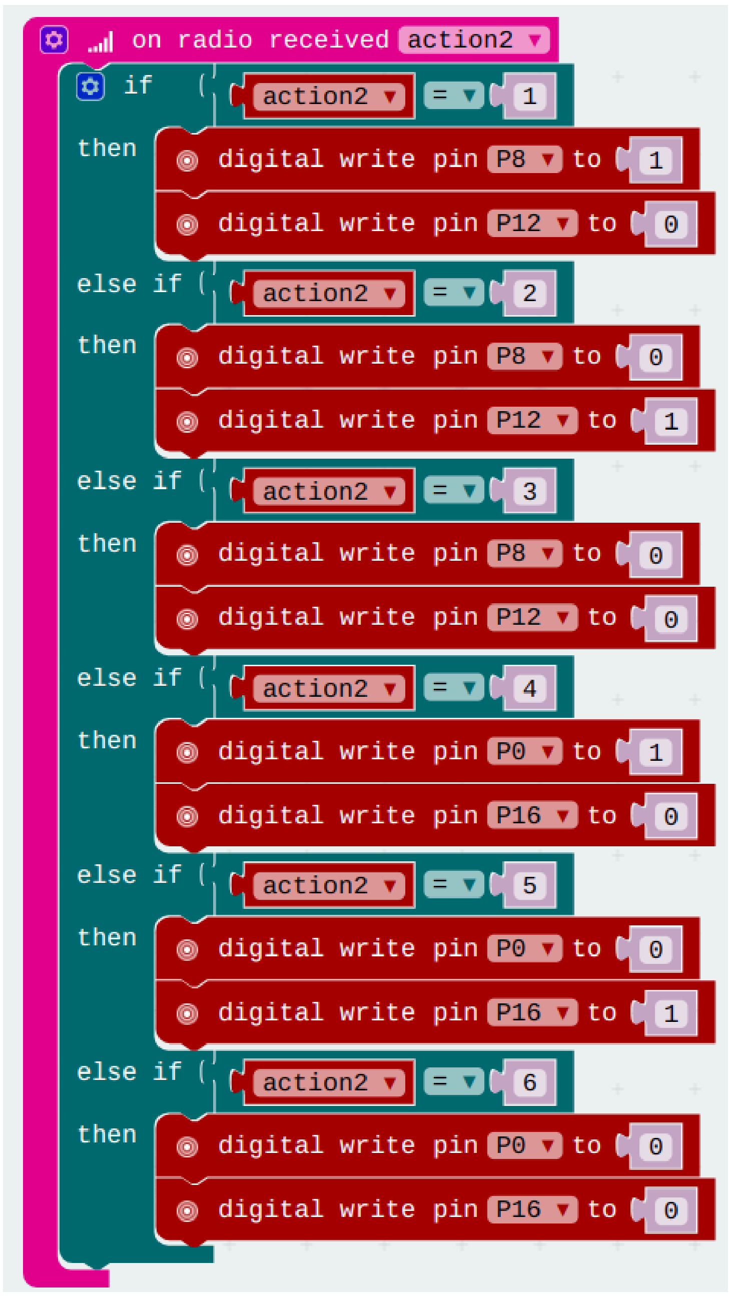
Motor 2 Connections Truth Table		
Pin 0	Pin 16	
0	0	Stay straight
1	0	Spin clockwise (turn right)
0	1	Spin anti clockwise (turn left)
1	1	Stay straight

Both 0, 0 and 1, 1 will cause the front wheels to stay straight but we will only use 0, 0

The order in which this happens

- micro:bit 2 receives a message from micro:bit 1 in the form of a number between 1 and 6
- micro:bit 2 uses that number to decide what instructions to send through the pins to the controller.
- The controller in turn sends power to the motors telling them to either stop or spin in one direction or the other particular direction. For one set of wheels that will make them go forward or backwards and for the other set of wheels that will make them turn left or right.

And the code to make it happen



```
on radio received action2
  if action2 = 1
  then
    digital write pin P8 to 1
    digital write pin P12 to 0
  else if action2 = 2
  then
    digital write pin P8 to 0
    digital write pin P12 to 1
  else if action2 = 3
  then
    digital write pin P8 to 0
    digital write pin P12 to 0
  else if action2 = 4
  then
    digital write pin P0 to 1
    digital write pin P16 to 0
  else if action2 = 5
  then
    digital write pin P0 to 0
    digital write pin P16 to 1
  else if action2 = 6
  then
    digital write pin P0 to 0
    digital write pin P16 to 0
```

It is easier logically to think of the two micro:bits using different code. One is programmed to send instructions to the other and the other is programmed to control the motors. In fact the way the code is written above, all of the code can be loaded on both micro:bits. This makes it easier if you are just using one computer as the makecode.microbit.org website doesn't play nice with the idea of have two tabs open with different code in each. Not an issue if you have a group of students who will programme each micro:bit using a separate computer.

Diagnosing problems

Problem
The car goes forward when it is meant to go backwards and backwards when it is meant to go forward.

Solution
The programming instructions from the micro:bit aren’t going to the correct motor connection. Either swap the cables going to the motor with each other OR swap the cables for that motor connecting to the pins on the edge connector (for L9110S only) OR change the code to swap which pins need to be turned on or off to go forwards and backwards - we suggest you swap the cable connections nearest the motor

Problem
The car turns left when it is meant to turn right and right when it is meant to turn left.

Solution
The programming instructions from the micro:bit aren’t going to the correct motor connection. Either swap the cables going to the motor with each other OR swap the cables for that motor connecting to the pins on the edge connector (For L9110S only) OR change the code to swap which pins need to be turned on or off to go left and right - we suggest you swap the cable connections nearest the motor.

Problem
Nothing happens

Solution
Check the batteries to make sure they aren’t drained of power. Test with new batteries if possible.
Check the cables are connected correctly.
Check the micro:bit isn’t connected to the edge connector upside down