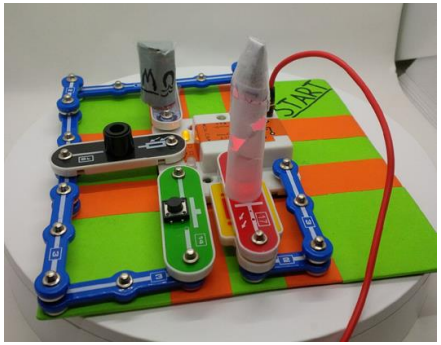
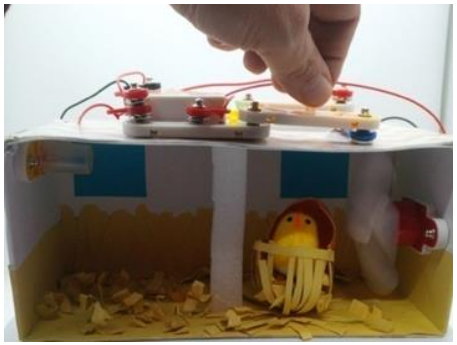
KODEKLIX®

Demo STEM Projects*

Resources: www.kodeklix.com/snapcpu4stem/



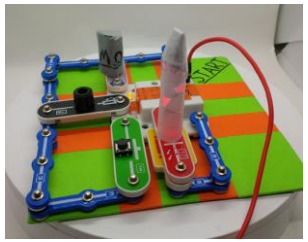


* See full manuals for details and product safety information

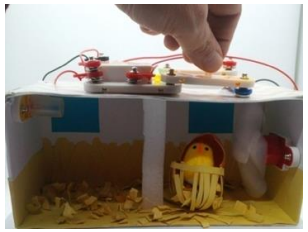
Project Overview

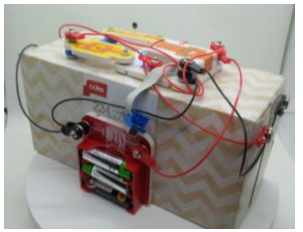
- Story-time with Lightning

- Board Game Fun



- Chicken Incubator





- Build a Model Appliance

Story-time with Lightning

Design Brief

Design an outback scene from Noongar culture which depicts their story telling with lightning and thunder using KodeKLIX.

Things to consider:

- Use a lamp to represent the lightning – perhaps hide the lamp in a cloud made of cotton wool
- Use a large inductive speaker for a deeper tone, rather than a small piezo speaker
- Experiment with BLOCKLY "note" values 128 and higher for making "white noise"

You will need to research about the Noongar people and their traditions and culture. Perhaps investigate the science behind what causes lightning and thunder. Also, check out what is "white noise".

Don't forget to document your finished project.

Note: this is a virtual project; no reason to hide under the bed as the lightning and thunder is not real!

Parts list (KodeKLIX electronics)

Lamp part #45, high power device [output]
Speaker [20] connected to the same high power device [output]
SnapCPU [for code]
Battery pack #70 [if portable power needed]
Baseplate to assist assembly
Links and wires

Parts list (Scene box)

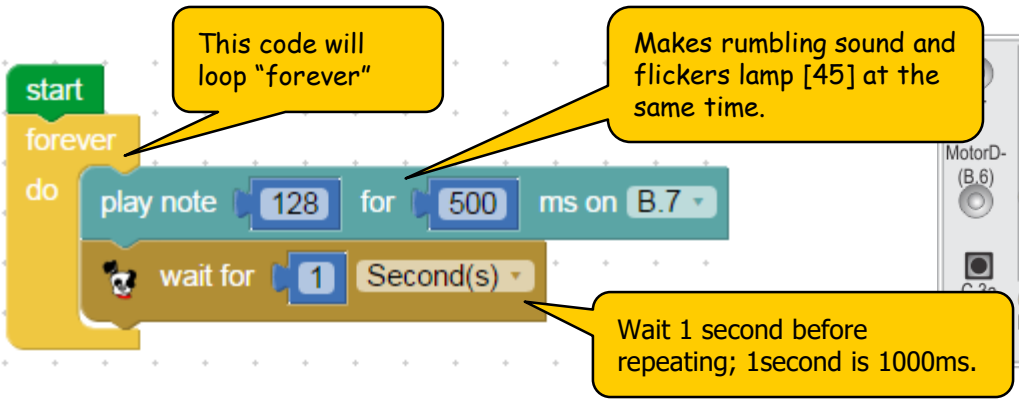
Shoe box
Paint for a background scene
Cotton wool to represent clouds and to hide the lamp
Other craft supplies

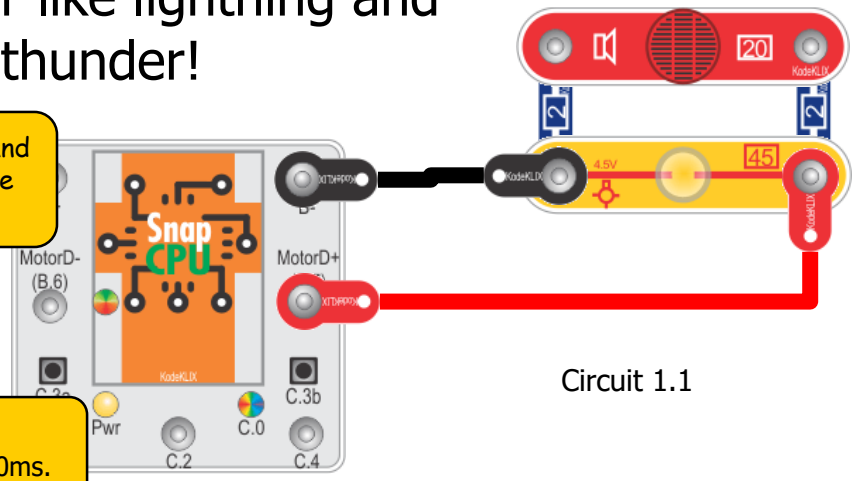
Completed ☐

Lightning & Thunder

Rumbling sound and flickering lamp

- Assemble the snap components as shown in Circuit 1.1
- Construct the following BLOCKLY code and download* the code to the SnapCPU™
- Observe the Lamp [45] flicker like lightning and the speaker [20] rumble like thunder!





Circuit 1.1

Challenge: modify the code to change blink rate of the lamp and help tell a story with mood and emotion.

* In order to download the coding changes, your SnapCPU™ will need to be powered-up and connected to a computer via its download link cable – refer download guide

* The SnapCPU™ is designed so that it can be directly connected to the terminals of the battery box and the polarity of the connections is correct.

Note: this will immediately power-up SnapCPU™ and start the code running.

Board Game Fun

Design Brief

Design a board game based on your favourite Book or Movie. Use KodeKLIX parts to make the game more exciting.

Things to consider:

- Use a simple layout for your game board where you move around the board clockwise or anti-clockwise
- Encourage interaction from the users; for example to collect things or count things
- Include a couple special "squares" where "special" things can happen; for example special sounds or actions
- Player tokens can be made "smart" by including a magnet in the base or a clear feature which glows when over a LED

Research how other game designers make their board games. Consider using the tune wizard to make a special tune and add mood, excitement or danger. Play your game with friends or family to test it out before finalising the design.

Advanced users: consider adding mechatronic features connected to motors or servos.

Parts list (KodeKLIX electronics)

Input sensors such as buttons [14], magnets [13] or LDR [16]
Speaker [20] or Piezo [11], connected high power device [output]
SnapCPU [for code]
Battery pack #70 [if portable power needed]
Baseplate to assist assembly
Links and wires

Optional parts:
Motors or Servos to add action, high power device [output]

Parts list (Game Board)

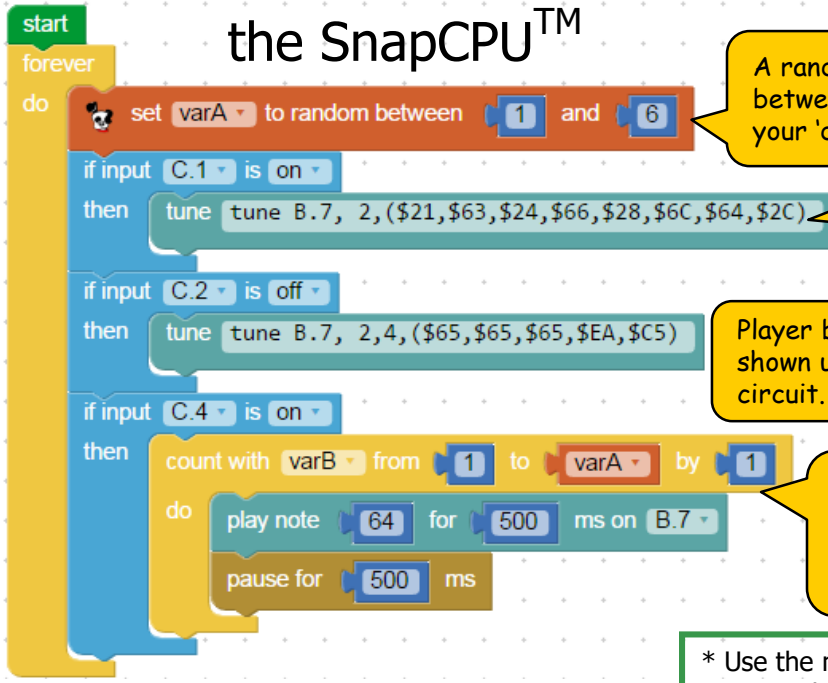
Thick cardboard
Paint or coloured paper for a background decorations
Markers for making labels on the board, instructions, etc
Other craft supplies

Completed ☐

Board Game (example design)

Board Game and Layout

- Assemble the snap components as shown in Circuit 1.2
- Construct the following BLOCKLY code and download the SnapCPU™

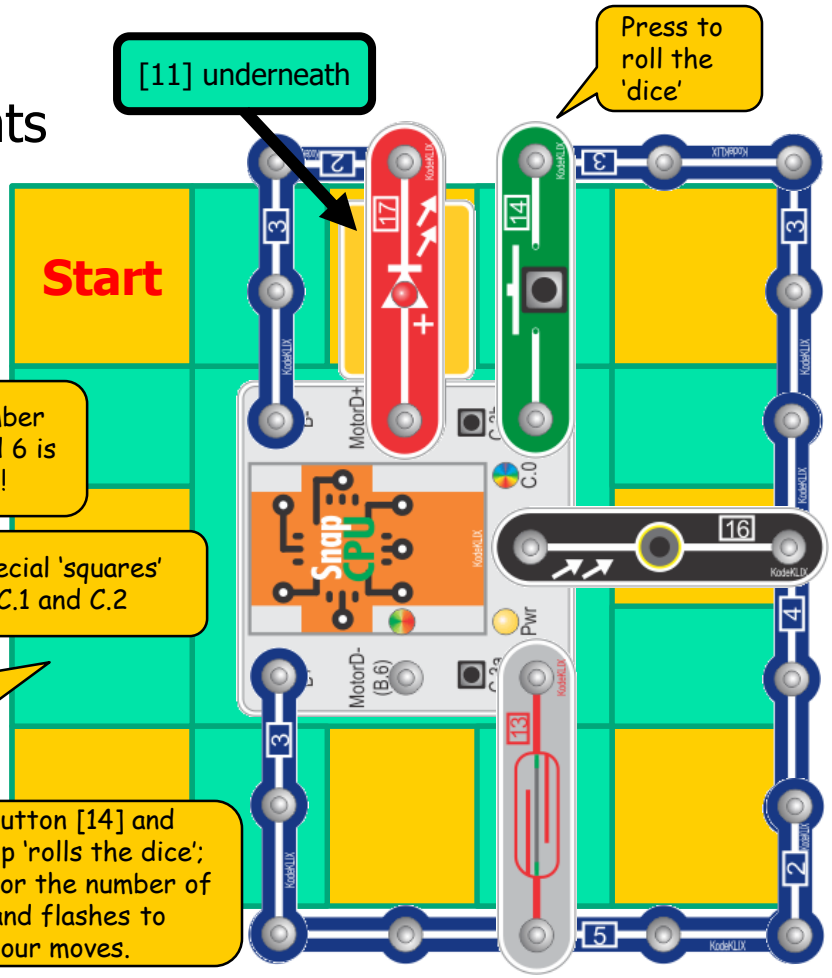


A random number between 1 and 6 is your 'dice' roll!

Special 'squares' at C.1 and C.2

Player board shown under circuit.

Press button [14] and this loop 'rolls the dice'; listen for the number of beeps and flashes to signal your moves.



Challenge: add a background and tokens

* Use the magnet to activate the reed sensor [13]. Activate the light sensor by making it dark. Magnets can be connected to the underside of your game tokens.

Circuit 1.2

Chicken Incubator

Design Brief

Design a model of an automatic chicken egg incubator using an electronic control system to monitor temperature using KodeKLIX parts. Things to consider:

- Egg environment to be kept at correct temperature
  - Heat using the incandescent lamp
  - Cool using a fan to blow air
- (extra coding credit) Alarm for over temperature
- (extra coding credit) Alarm for under temperature
- (extra coding credit) Alarms with status indicator using built-in RGB LED

You will need to research about chickens to find out what temperature is needed for the eggs to hatch and not get cooked, and how long it will take for the "eggs" to hatch. Don't forget to document your finished project.

Note: this is a virtual project; no chickens or eggs will be harmed in the experiment!

Parts list (KodeKLIX electronics)

Thermistor for measuring temperature #33T [input]
Lamp part #45, high power device [output]
Motor #39 with fan attachment, high power device [output]
SnapCPU [for code]
Piezo speaker #11 [if implementing alarm]
Battery pack #70 [if portable power needed]
Baseplate to assist assembly
Links and wires

Parts list (Incubator box)

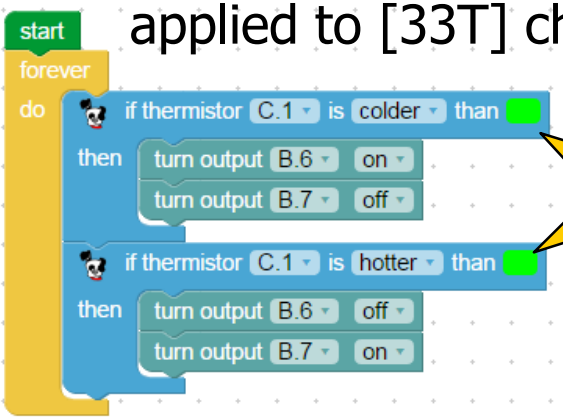
Shoe box
Insulation materials like straw or cotton
Other craft supplies

Completed ☐

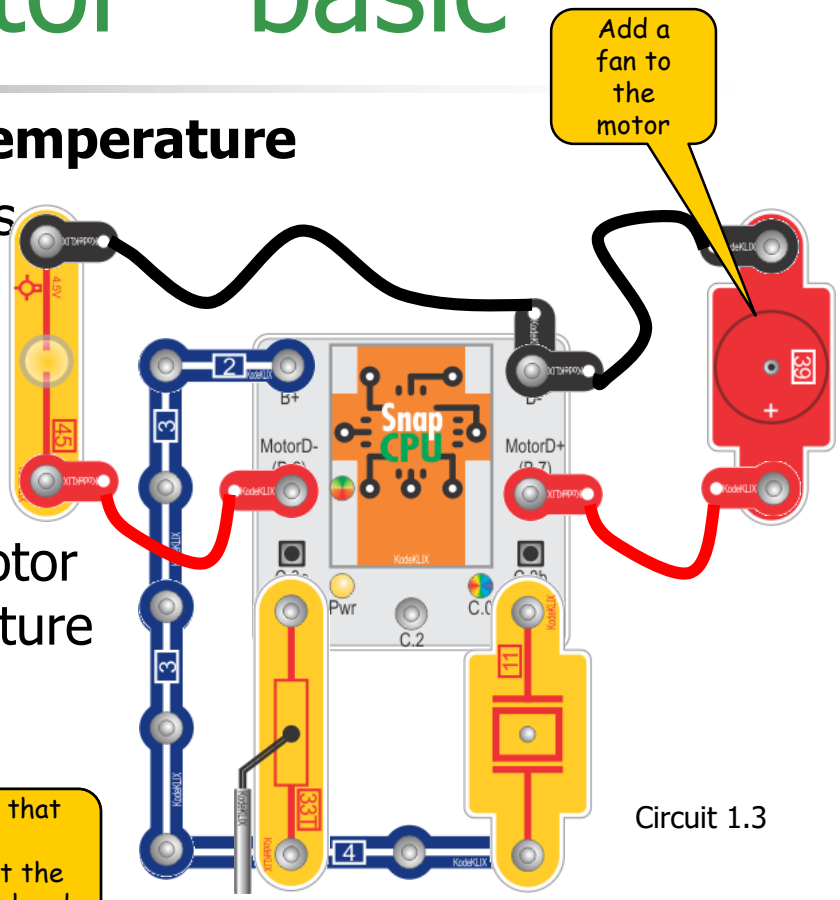
Chicken Incubator - basic

Motor & Lamp controlled by temperature

- Assemble the snap components as shown in Circuit 1.3
- Construct the following BLOCKLY code and download to the SnapCPU™
- Observe the Lamp [45] and motor toggle state when the temperature applied to [33T] changes...



Adjust this value so that the lamp and motor toggle when you heat the sensor [33T] in your hand



Circuit 1.3

Challenge: see next page

* In order to download the coding changes, your SnapCPU™ will need to be powered-up and connected to a computer via its download link cable – refer download guide

* The SnapCPU™ is designed so that it can be directly connected to the terminals of the battery box and the polarity of the connections is correct.

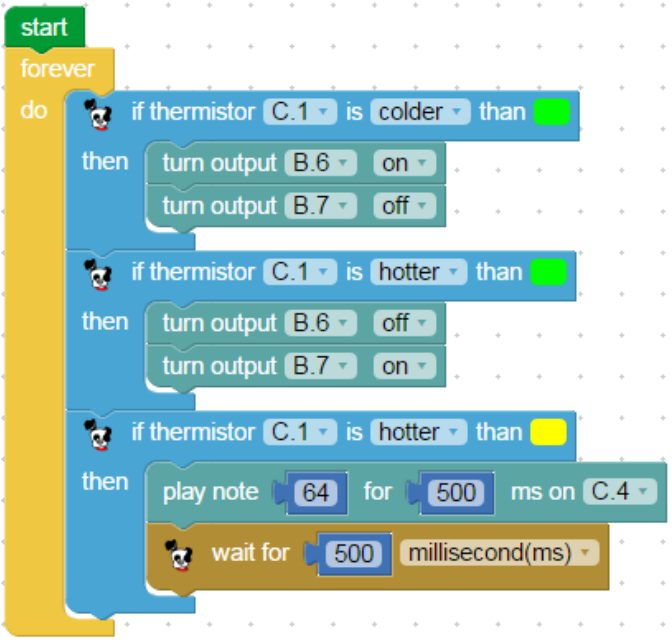
Note: this will immediately power-up SnapCPU™ and start the code running.

Chicken Incubator - advance

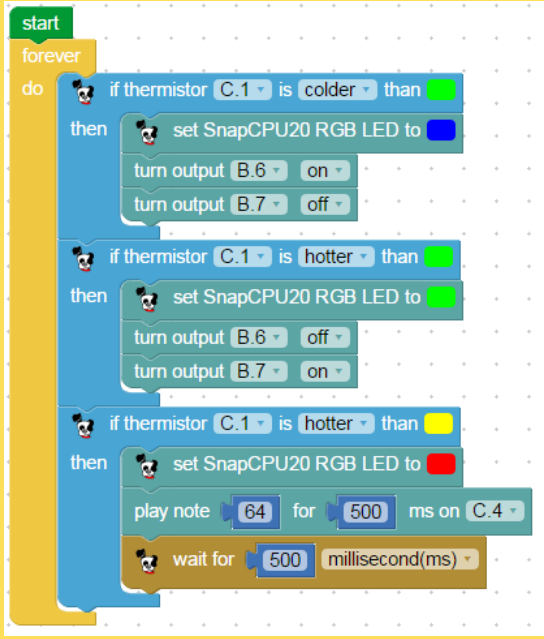
Motor & Lamp controlled by temperature, with alarms

- Assemble the snap components as shown in Circuit 1.3
- Construct each of the following BLOCKLY code and download to the SnapCPU™ to test the extra functionality…

Code for adding an over temperature alarm



Code for adding an over temperature alarm with LEDs



Build a Model Appliance

Design Brief

Modern washing machines are complicated machines which typically include a control computer. Use KodeKLIX parts to build a fully working model of this appliance.

Things to consider:

- Use a motor or servo to bring action into your appliance; eg the movement of the washing tub
- Use switches to control the operating mode. Modes can change speed or duration of the action
- Investigate the use of safety features to protect users of your appliance from danger

Research how safety is designed into products to protect users. Things like sensors on doors prevent machines from working until the doors are closed and safe. Imagine how dangerous it would be if you could use the microwave with the door open!

Note: There are many other appliances in your house or school. Perhaps for your next modelling project, try making a model of an air condition, microwave over, blender or other device using different input sensors!

Parts list (KodeKLIX electronics)

Motor or Servo, high power device [output]
Buttons for input; try using the coloured buttons [inputs]
Piezo Speaker [11] low or high power device [output]
SnapCPU [for code]
Battery pack #70 [if portable power needed]
Baseplate to assist assembly
Links and wires

Note: Servos must connect to B.2 (same as B.6 snap) or B.3 (same as B.7 snap) to work. Servos need to be initialised when first used (see code example)

Parts list (Appliance Model)

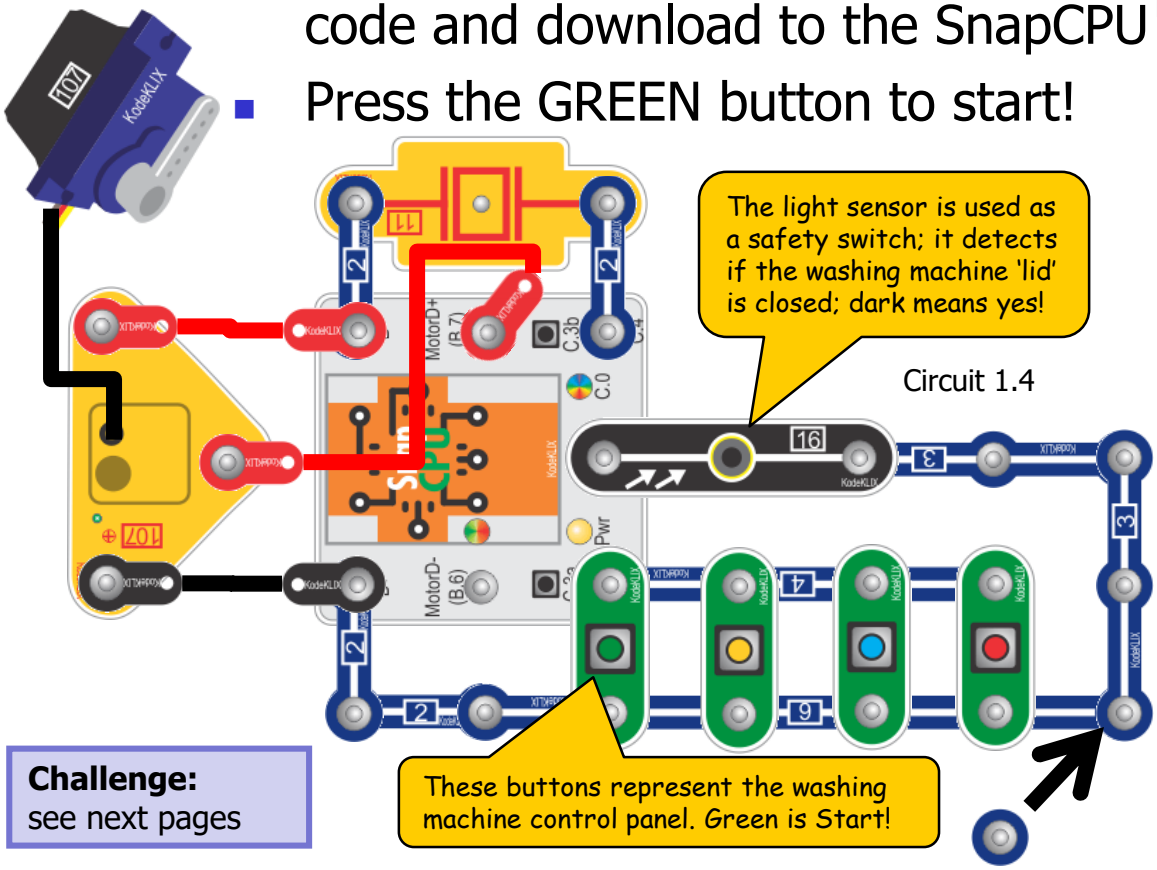
Cardboard for constructing the machine case and lid
A plastic tub eg small yogurt container
Paint for decoration scene
Other craft supplies

Completed ☐

Washing Machine - basic

Using a Servo* Motor to simulate the washing tub 'swish'

- Assemble the snap components as shown in Circuit 1.4
- Construct the following BLOCKLY code and download to the SnapCPU™
- Press the GREEN button to start!



Circuit 1.4

Challenge:
see next pages

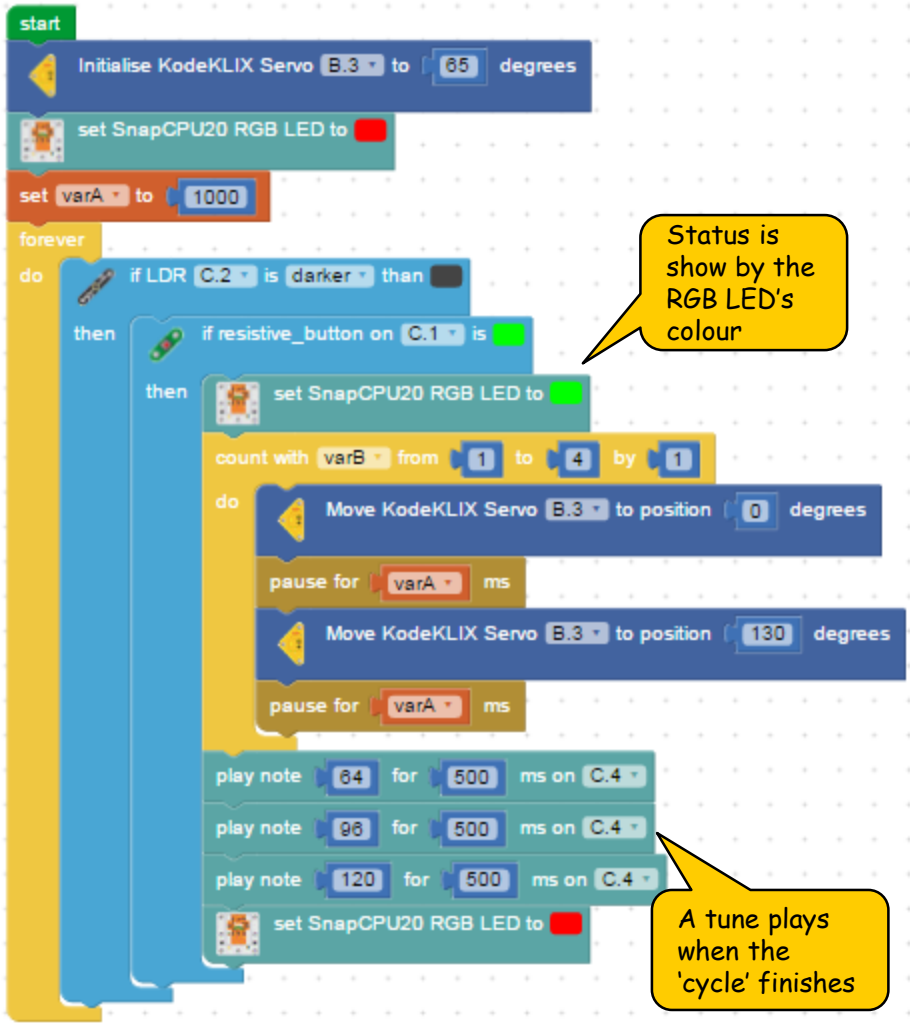
* The servo is a motor which moves to a requested angular position. The servo needs three wires: battery power + and -, as well as the control signal that tells it what position to move to and wait.

Completed ☐

Washing Machine - advance

Adding a safety switch

- Assemble the snap components as shown in Circuit 1.4
- The BLOCKLY code can be updated to include a check for the lid being closed
- Download to the SnapCPU™ to test that the machine only starts moving when the safety 'lid' is closed



Challenge: integrate the electrical parts into a model of the washing machine made with construction materials using a tub connected to the servo, a safety 'lid' and control panel. You can move the light sensor [16] using snap wires.

Washing Machine - deluxe

Adding mode switches for a control panel

- Assemble the snap components as shown in Circuit 1.4
- The BLOCKLY code can be further updated to include a checks for which button is pressed to determine the 'mode'
- Download to the SnapCPU™ to test that the machine only starts when the 'lid' is closed, and runs different 'modes'

Challenge: the coloured buttons can all be connected to the one input snap. This is called multiplexing.

Add code for the red button and create a special function for it!

Challenge yourself to build models of other items in your house or school.