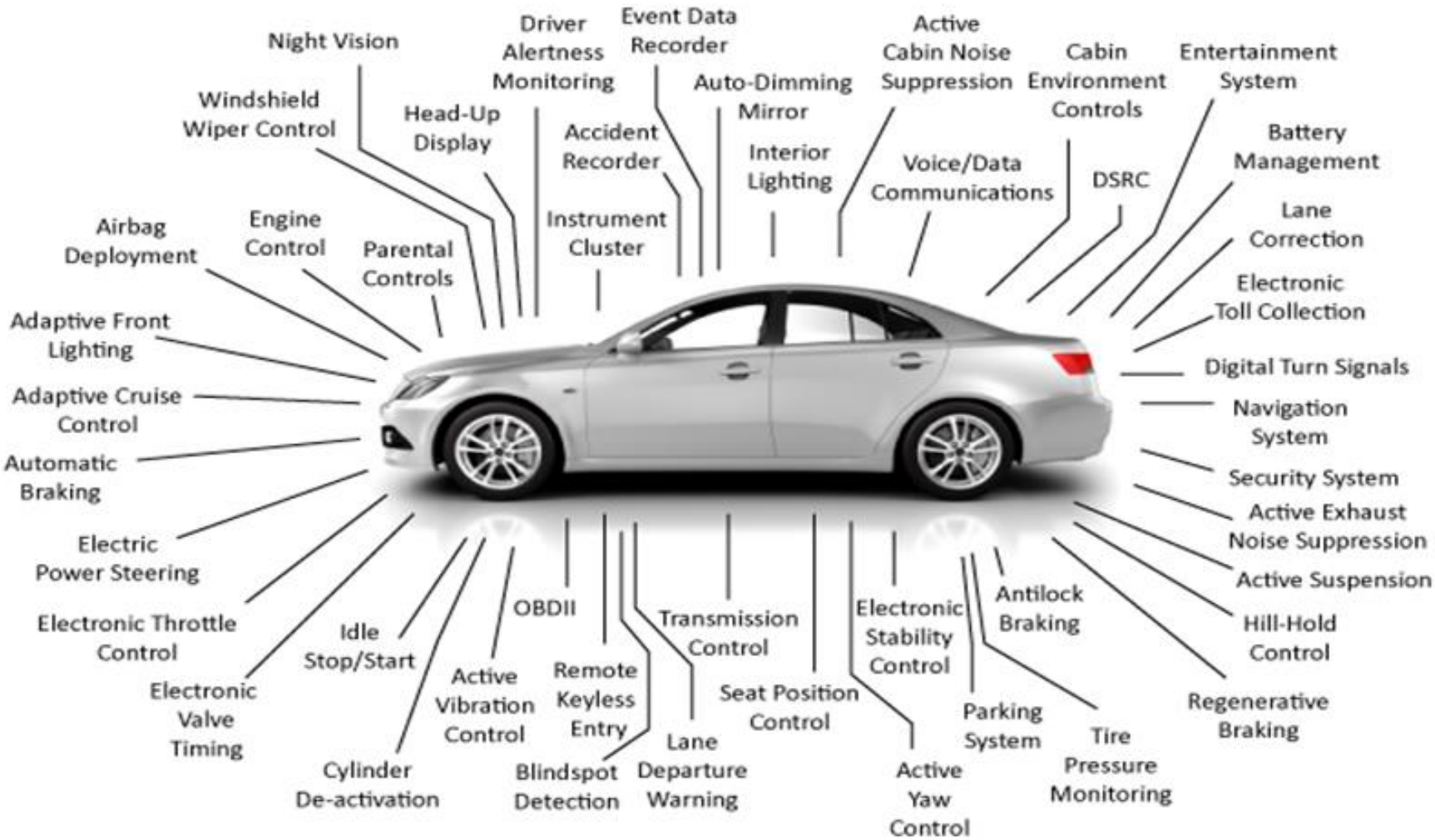
Modern Automobile Controls

SnapCPU Project Guide

Modern Automobile Controls

A modern vehicle has hundreds of computer controlled functions



Automobile Control Projects

Powertrain
- Battery / Solar (for motor)
- Acceleration (incl. assisted)

Automatic Climate Control (Aircon)
- Turn on fan when over-temperature

Digital Entertainment
- Play tunes from stored memory

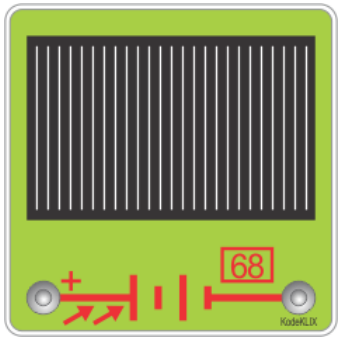
Vehicle Security
- Bump or Tilt detection alarm

Automatic Headlights
- Light them up in dim conditions

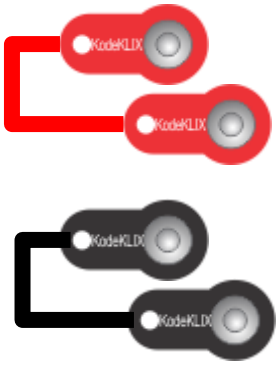
Extra* Components Needed

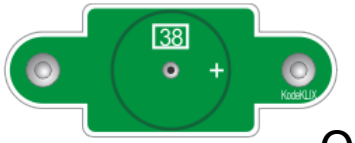
OR

2x required

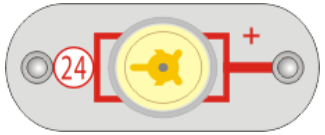


1x required



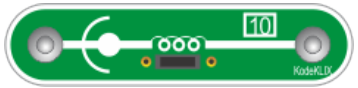


OR

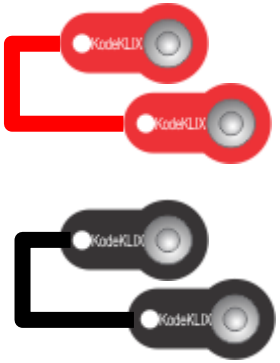


1x required

1x required



1x required



4x required

* Additional components not in the SnapCPU starter kit

Vehicle Security

Completed

Skill needed

PROJECT SCOPE:

- Use a tilt or vibration sensor to turn on the alarm if the vehicle is bumped
  - The alarm should only be active when the car is switched off; include a switch to represent the key and blink C.0 to signal
  - When triggered, sound the alarm for 1min
- Challenge: switch alarm off when 'key ON'

Hint: when using vibration or tilt sensors the 'trigger' could be very brief. You will need to continually check the status of the sensor!

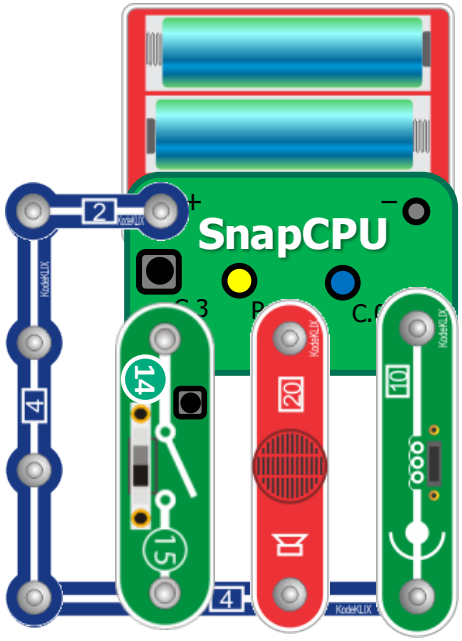
Vehicle Security

Completed

Skill needed

MONITOR FOR VIBRATION OR TILT, AND TRIGGER ALARM

- Assemble the snap components as shown
- Construct BLOCKLY code to check when the slide switch (key) is "off" and to then flash the C.0 LED
- Construct BLOCKLY code to check for the slide switch (key) being off and the vibration / tilt switch being triggered
  - If triggered make an alarm sound for 60s
  - If you want an extra challenge, use the 'key' to reset the alarm when it is switched to the ON position



Hint #1: Use two separate coding blocks / tasks.

Hint #2: To reset code, use this block: restart all tasks

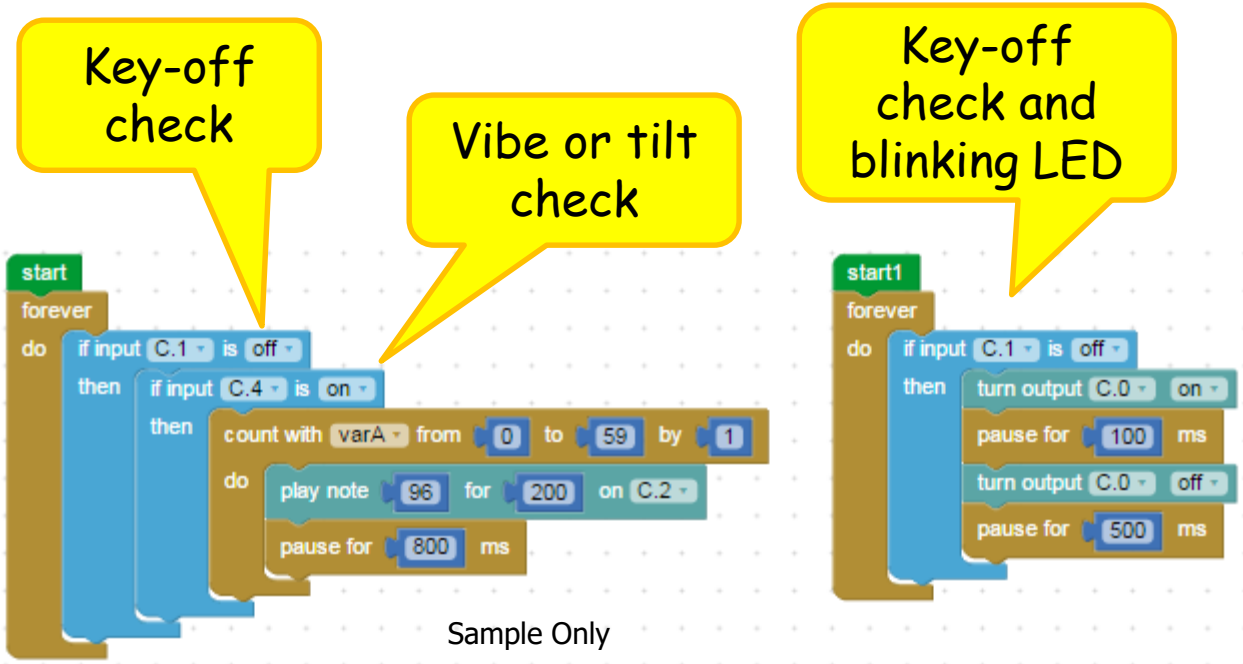
Vehicle Security #1

Completed ☐

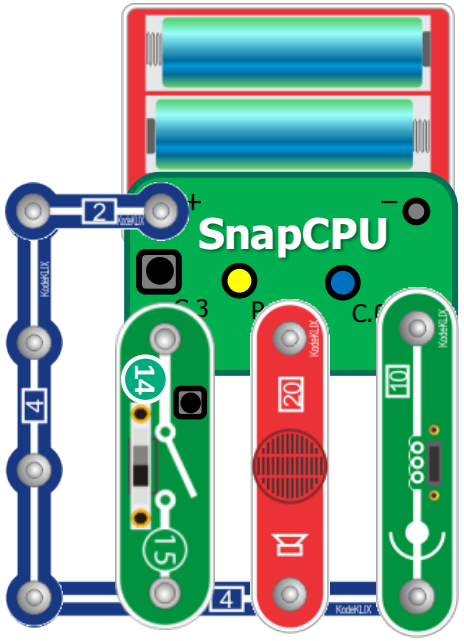
Skill needed

MONITOR FOR VIBRATION OR TILT, AND TRIGGER ALARM

- Assemble the snap components as shown
- Construct the BLOCKLY code below and adjust the sounds played to suit



Sample Only



Note: with this code, once the alarm is triggered it will not stop until 1 minute (60 beeps) passes

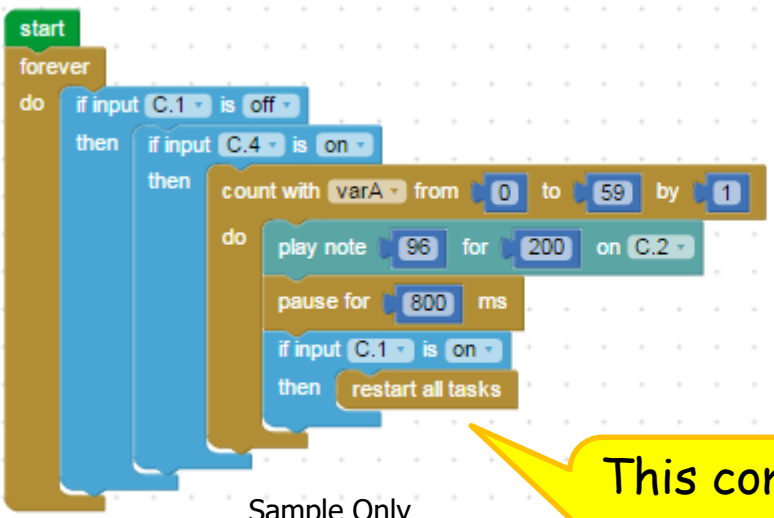
Vehicle Security #2

Completed ☐

Skill needed

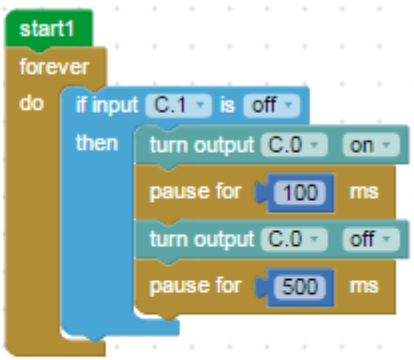
MONITOR FOR VIBRATION OR TILT, AND TRIGGER ALARM

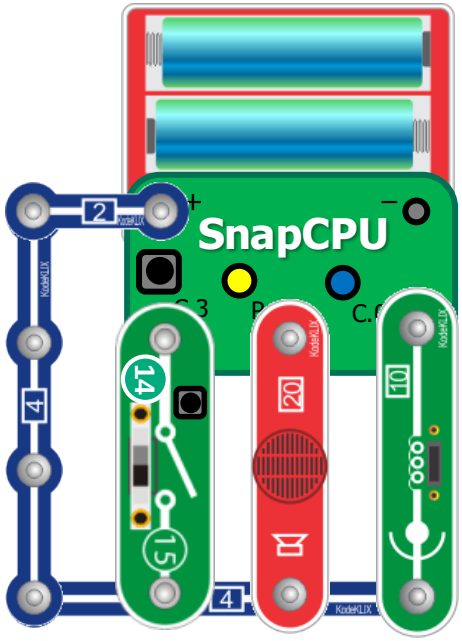
- Assemble the snap components as shown
- Construct the BLOCKLY code below and adjust the sounds played to suit



Sample Only

This command restarts the SnapCPU!





Note: this code resets if C.1 is switched ON because of the check within the alarm loop

Automatic Headlights

Completed

Skill needed

PROJECT SCOPE:

- Use a light sensor to decide when to operate the 'headlights' automatically
  - When bright or day, switch off
  - When getting dark, switch on
- The light sensor is a LDR (light dependent resistor). You will need to experiment to get the values in your code 'just right'

Hint: overlap the on/off logic so that the headlights don't get stuck toggling on/off

Automatic Headlights

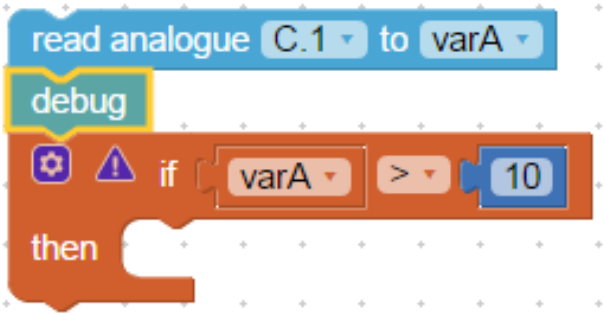
Completed ☐

Skill needed

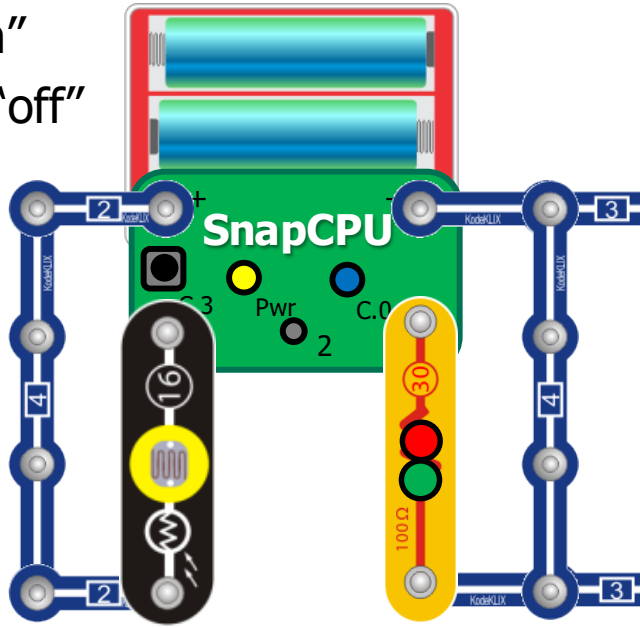
DETECT THE LEVEL OF AMBIENT LIGHT AND USE THAT INFORMATION TO SWITCH HEADLIGHTS ON/OFF

- Assemble the snap components as shown
- Construct BLOCKLY code
  - Code will need to read the analogue value

  - Use one **IF…varA…THEN** to switch "on"
  - Use second **IF..varA…THEN** to switch "off" (this is called hysteresis)



Hint: Use DEBUG to see what the sensor value is reading and use that information to adjust the code values



Automatic Headlights

Completed ☐

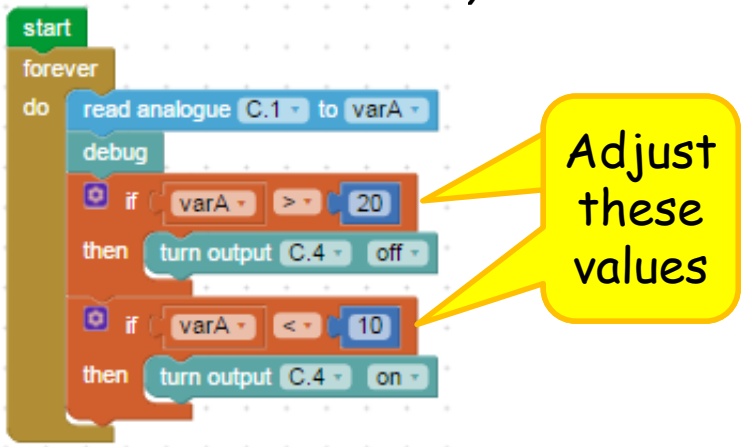
Skill needed

DETECT THE LEVEL OF AMBIENT LIGHT AND USE THAT INFORMATION TO SWITCH HEADLIGHTS ON/OFF

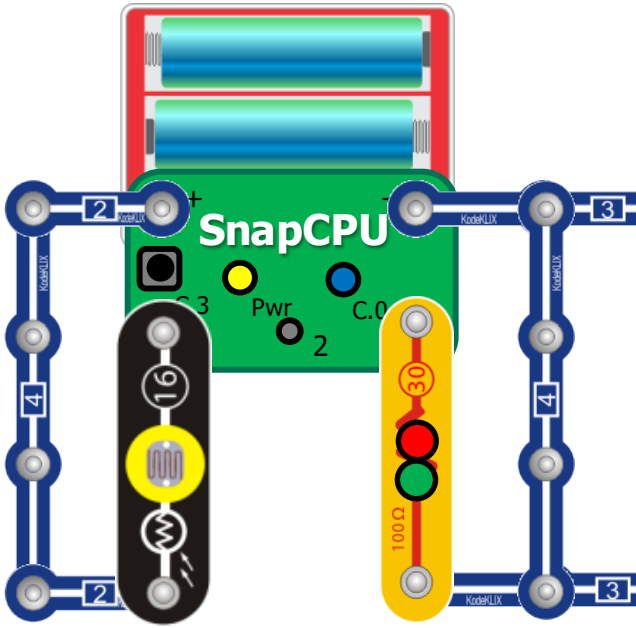
- Assemble the snap components as shown
- Construct the BLOCKLY code shown
  - Adjust the values of the constants used in the if then statement to switch on/off as required
  - The values should not overlap (this is called hysteresis)

Hint: Use DEBUG to see what the sensor value is reading and use that information to adjust the code values

Sample Only



Code: Automatic Headlights



Digital Entertainment

Completed

Skill needed

PROJECT SCOPE:

- Play one of three tunes when buttons connected to C.1, C.3, or C.4 are pressed
  - Tunes can be either built into the SnapCPU or downloaded
- Extra Challenge:
  - Use a 'long press' of each button to select a second set of three tunes (2x3 =6 songs)

Note: once a tune starts playing, other button presses are ignored…

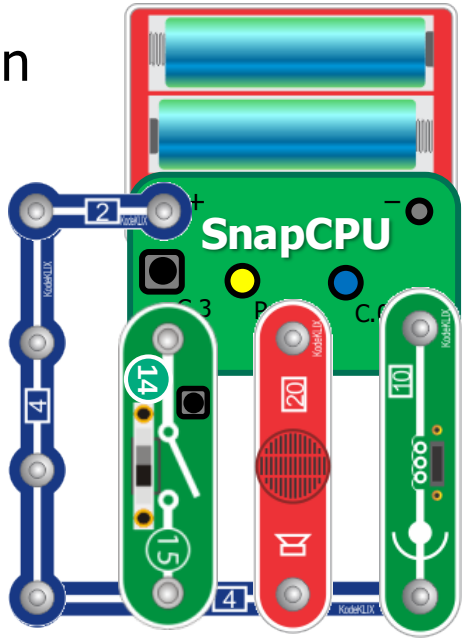
Digital Entertainment

Completed

Skill needed

USER SELECTION OF TUNE, PLAYED THROUGH SPEAKER

- Assemble the snap components as shown
- Construct BLOCKLY code to play music when one of the three buttons is pressed
- Music can be any of the following:
  - Sounds
  - Tunes
  - In-built PLAY songs
- Challenge 1:
  - Three tune player, one for each button
- Challenge 2:
  - Six tune player, two for each button
  - Hold button for approx 1second to start second tune



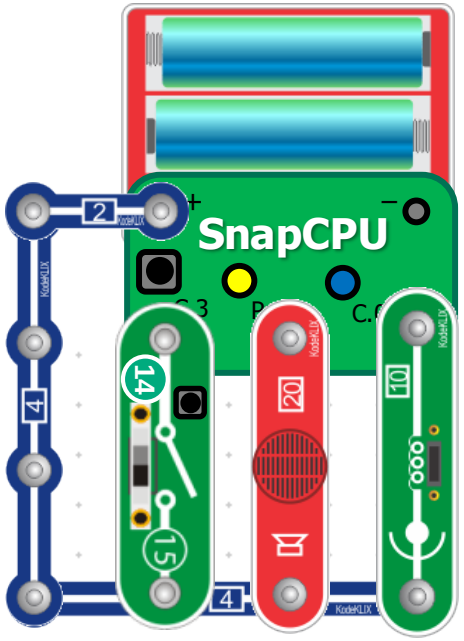
Digital Entertainment #1

Completed ☐

Skill needed

USER SELECTION OF TUNE, PLAYED THROUGH SPEAKER

- Assemble the snap components as shown
- Construct the BLOCKLY code shown; below is the three tune player



Sample Only

```
start
forever
do
  if input C.1 is on
  then play Happy Birthday on C.2
  if input C.3 is on
  then play Jingle Bells on C.2
  if input C.4 is on
  then tune tune 0, 4,($AA,$85,$43,$42,$40,$8A,$C5,$43,$42,$40,$8A,$C5,$43,$42,$43,$C0)
```

Adjust these tunes

Code: Digital Entertainment System #1

Digital Entertainment #2

Completed ☐

Skill needed

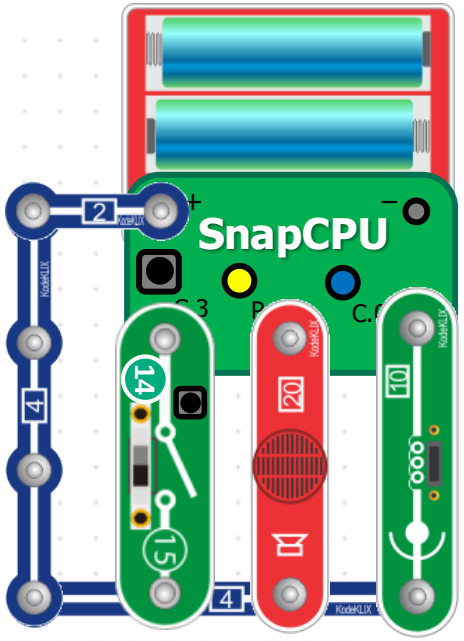
USER SELECTION OF TUNE, PLAYED THROUGH SPEAKER

- Assemble the snap components as shown
- Construct the BLOCKLY code shown; below is the six tune player

Sample Only

```
start
forever
do
  if input C.1 is on
  then
    pause for 1000 ms
    if input C.1 is off
    then play Happy Birthday on C.2
    else play Silent Night on C.2
  if input C.3 is on
  then
    pause for 1000 ms
    if input C.3 is off
    then play Jingle Bells on C.2
    else play Rudolph on C.2
  if input C.4 is on
  then
    pause for 1000 ms
    if input C.4 is off
    then tune tune 0, 4,($AA,$85,$43,$42,$40,$8A,$C5,$43,$42,$40,$8A,$C5,$43,$42,$43,$C0)
    else tune tune 0, 2,($21,$63,$24,$66,$28,$6C,$64,$2C,$27,$6C,$63,$2C,$26,$64,$2C,$21,$63,$24,$66,$41,$2C,$C0)
```

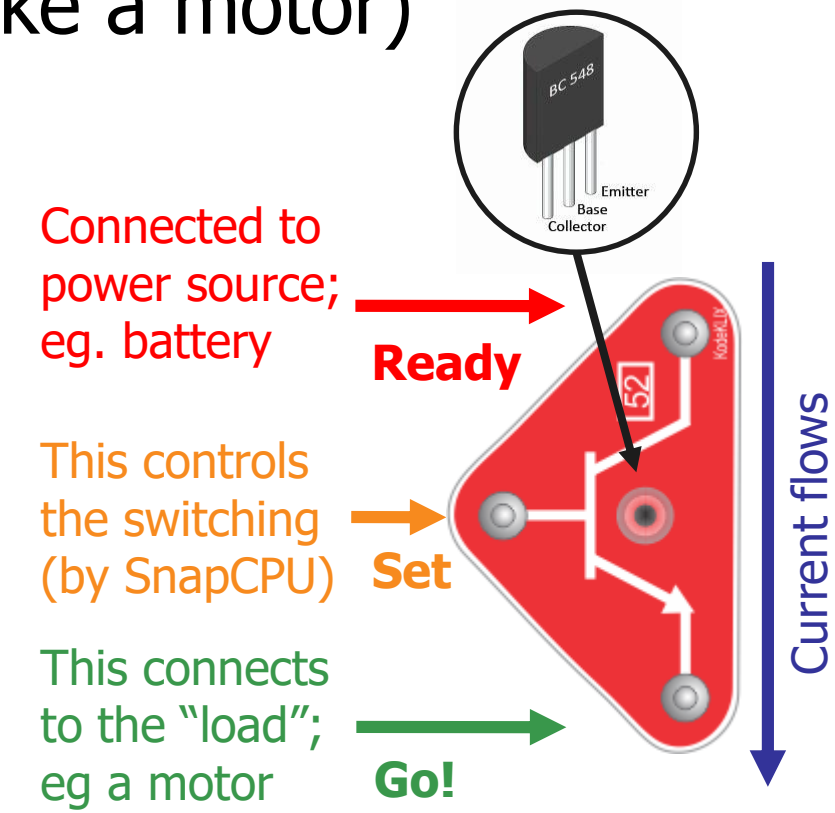
Adjust these tunes



Code: Digital Entertainment System #2

Introducing the Transistor

- A transistor is a switch which needs a little energy to turn things ON/OFF that require a lot of electricity (like a motor)
- Transistors are easy to use if you understand the way electricity is connected to the motor, etc



KodeKLIX® transistors components are modified internally so as to respond without the need for additional external components. The connections Ready, Set, Go! have technical names in the electronic industry like Collector, Base and Emitter (or Source, Gate, Drain for a more efficient FET device).

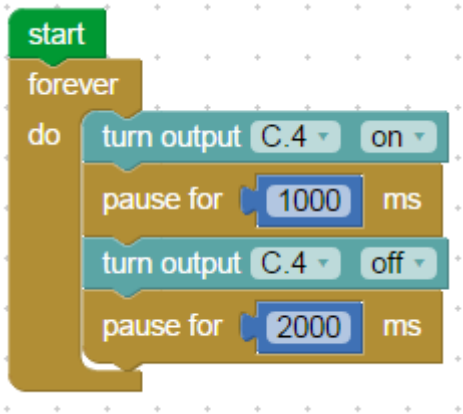
Simple Transistor Circuit

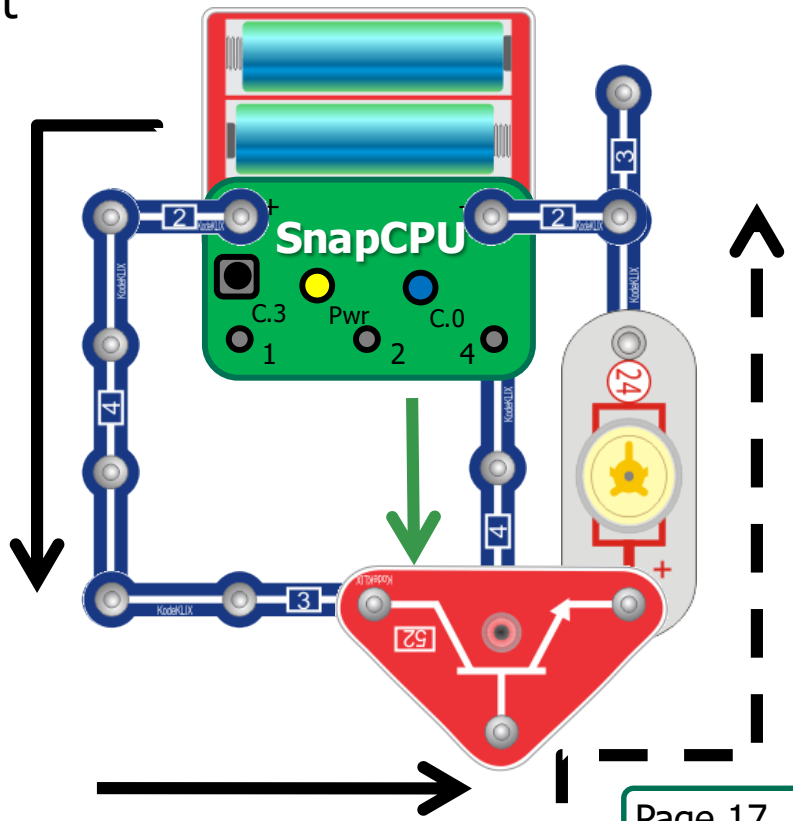
Completed ☐

Skill needed

SnapCPU CONTROLLED MOTOR SWITCH

- Assemble the snap components as shown
- Construct BLOCKLY code as shown
  - Transistor will allow the electricity to flow to the motor only if output C.4 is switched ON





Completed

Automatic Climate Control

Skill needed

PROJECT SCOPE:

- Use a temperature sensor to decide when to operate the fan motor
  - Too cold, switch off
  - Too hot, switch on

  Hint: overlap the on/off logic so that the motor doesn't get stuck toggling on/off
- The temperature sensor is a thermistor (temperature sensitive resistor). You will need to experiment to get the values in your code 'just right'

Completed ☐

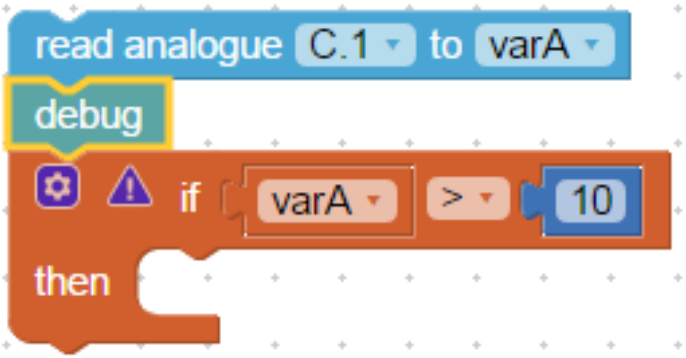
Automatic Climate Control

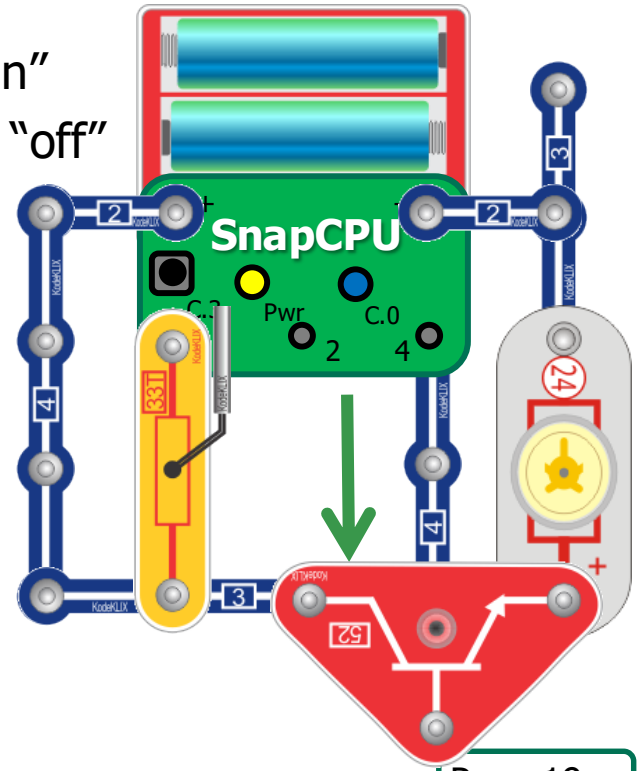
Skill needed

TEMPERATURE USED TO SWITCH AirCon MOTOR ON/OFF

- Assemble the snap components as shown
- Construct BLOCKLY code
  - Code will need to read the analogue value from the sensor

  - Use one **IF…varA…THEN** to switch "on"
  - Use second **IF..varA…THEN** to switch "off" (this is called hysteresis)





Hint: Use DEBUG to see what the sensor value is reading and use that information to adjust the code values

Completed ☐

Automatic Climate Control

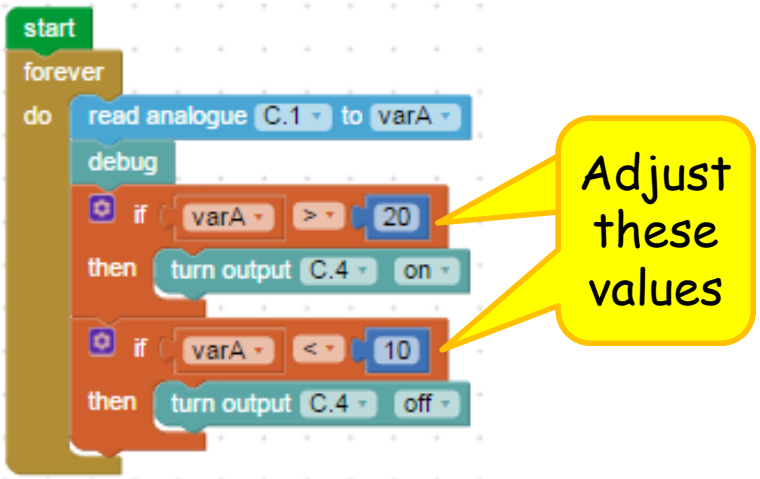
Skill needed

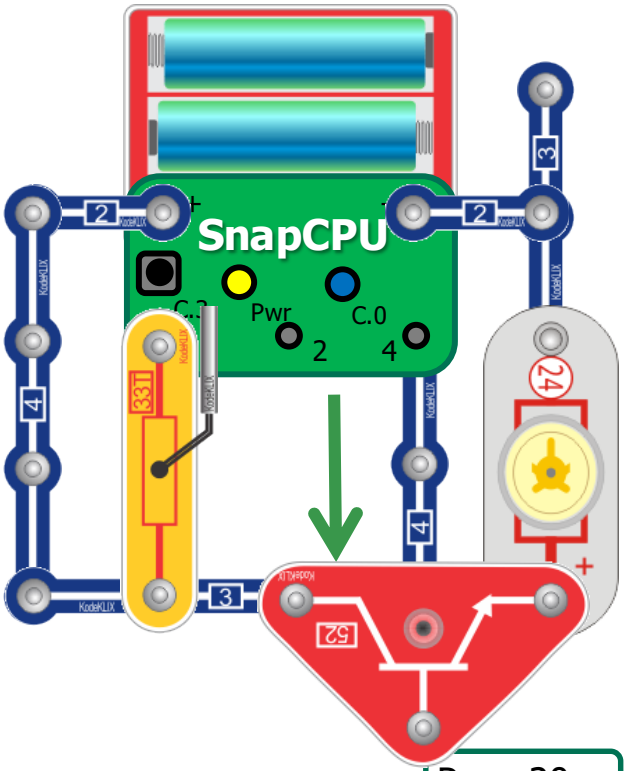
TEMPERATURE USED TO SWITCH AirCon MOTOR ON/OFF

- Assemble the snap components as shown
- Construct the BLOCKLY code shown
  - Adjust the values of the constants used in the if then statement to switch on/off as required
  - The values should not overlap (this is called hysteresis)

Hint: Use DEBUG to see what the sensor value is reading and use that information to adjust the code values

Sample Only





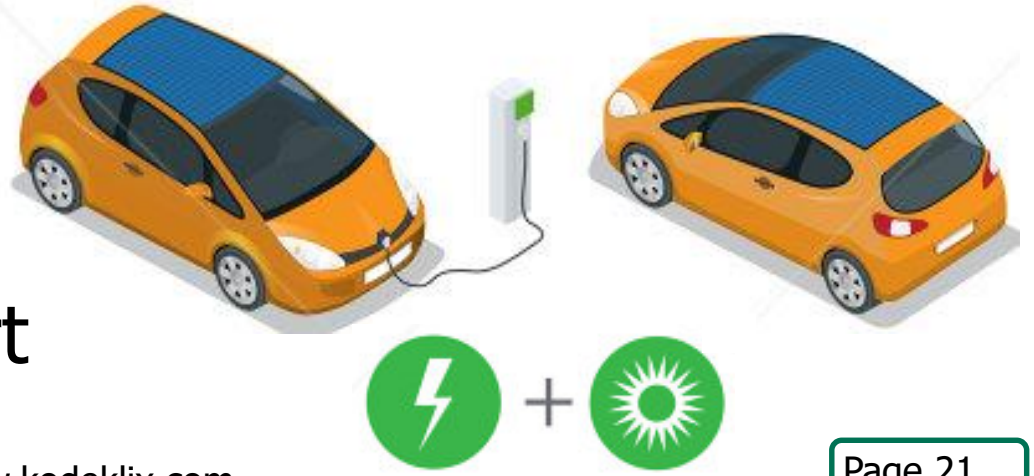
Completed

Powertrain: Battery+Solar

Skill needed

PROJECT SCOPE:

- Only operate powertrain motor when pedal switch (C.3) is pressed
- Switch between solar and battery automatically (eg if going into a tunnel)
  - If you have enough sun, run from solar
  - If not, run motor from battery
- Use battery to help launch vehicle from a standing start



Powertrain: Battery

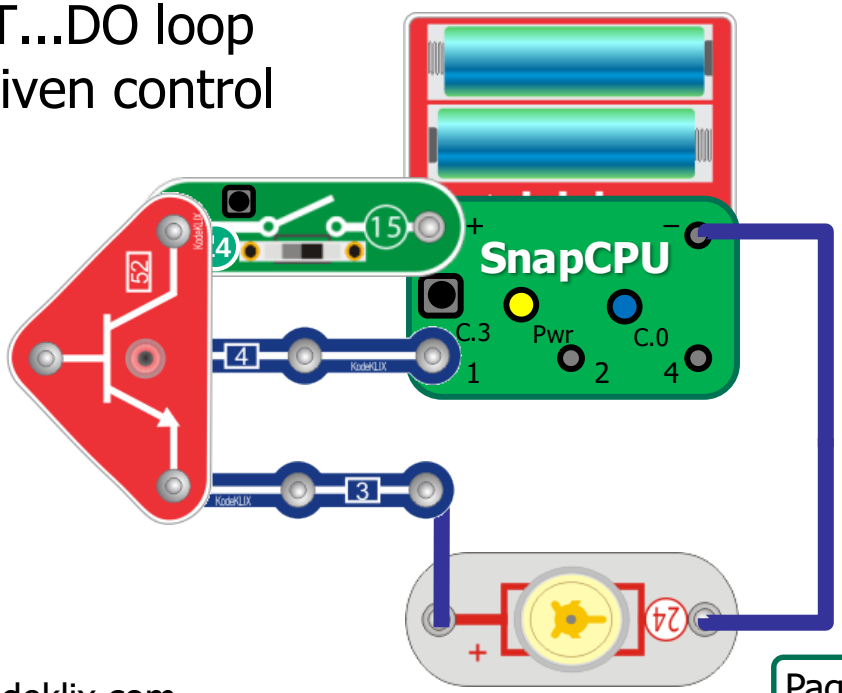
Completed

Skill needed

DRIVE MOTOR POWERED BY BATTERY

- Assemble the snap components as shown for "part 1"
  - The slide switch isolates battery power
- Construct BLOCKLY code:
  - You will need to read the pedal switch C.3
  - If pressed, activate the battery by switching output C.1 "ON"
- Using the WHILE...INPUT...DO loop allows for better state driven control

- Test that the transistor works as required



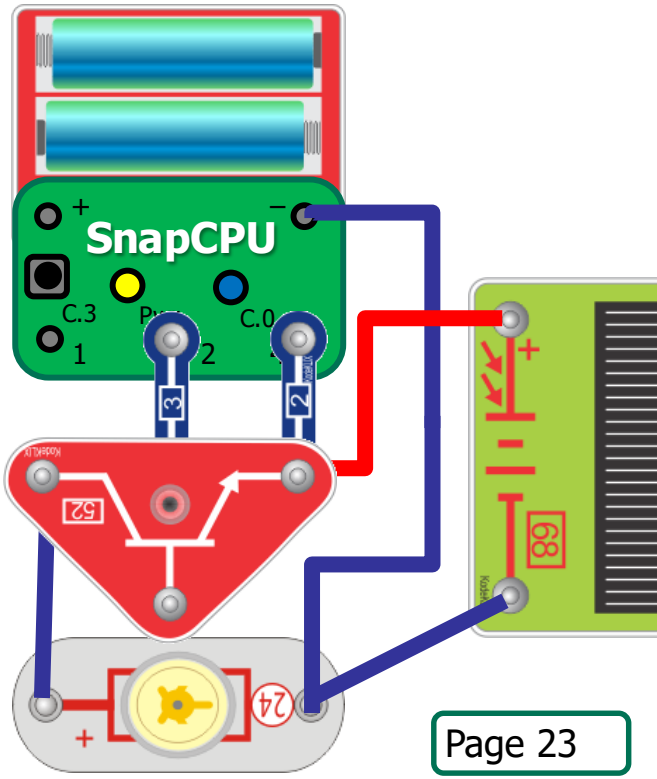
Powertrain: Solar

Completed

Skill needed

DRIVE MOTOR POWERED BY SOLAR ONLY WHEN SUFFICIENT LIGHT IS AVAILABLE

- Assemble the snap components as shown for "part 2"
- Construct BLOCKLY code:
  - You will need to read the pedal switch C.3
  - If pressed, activate the alternate solar circuit by switching output C.2 "ON"
  - The solar cell is connected to input C.4 so your code can check what solar power is available (by reading its voltage signal)



Note: the second power source is the solar cell. This only powers the motor circuit. The battery is still needed to power the SnapCPU.

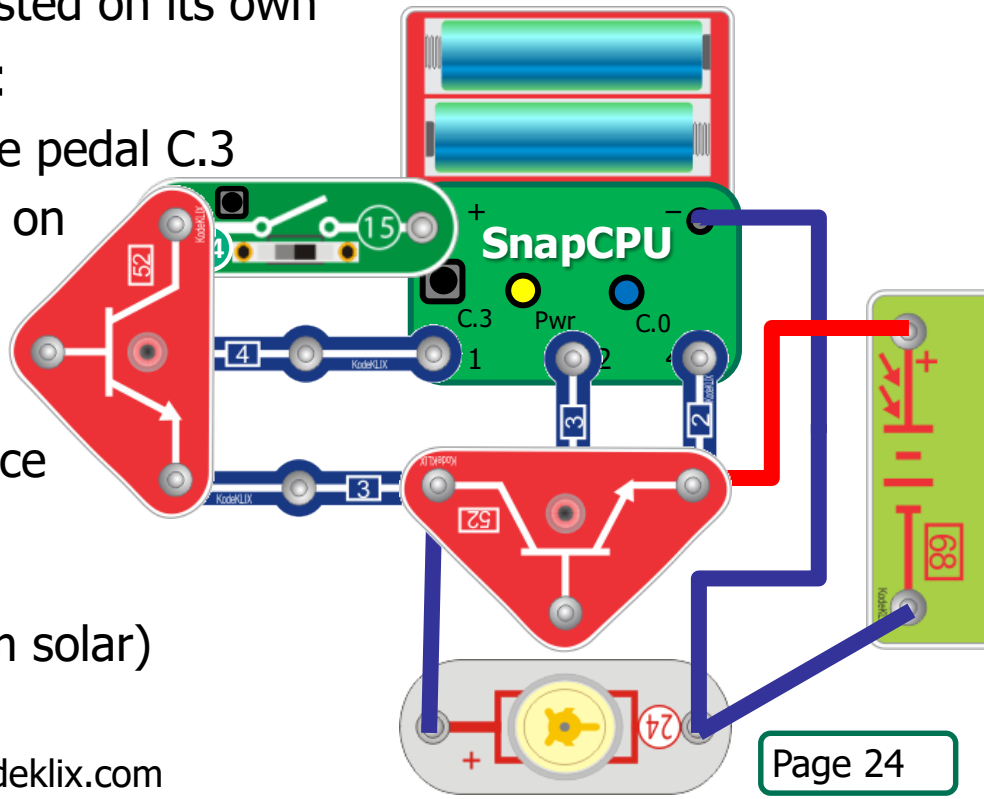
Completed

Powertrain: Battery+Solar

Skill needed

CONNECT BOTH BATTERY and SOLAR CIRCUITS

- Assemble the snap components as shown for "part 1"; and then merge in "park 2" into the circuit design
  - Connect the solar panel (use flex wires if needed)
  - The slide switch isolates battery power allowing the solar circuit to be tested on its own
- Construct BLOCKLY code:
  - You will need to read the pedal C.3
  - Sense C.4 for a decision on the status of the solar source
  - Control C.1 and C.2 to select which power source
  - Add a status signal using LED C.0 (ON means running from solar)



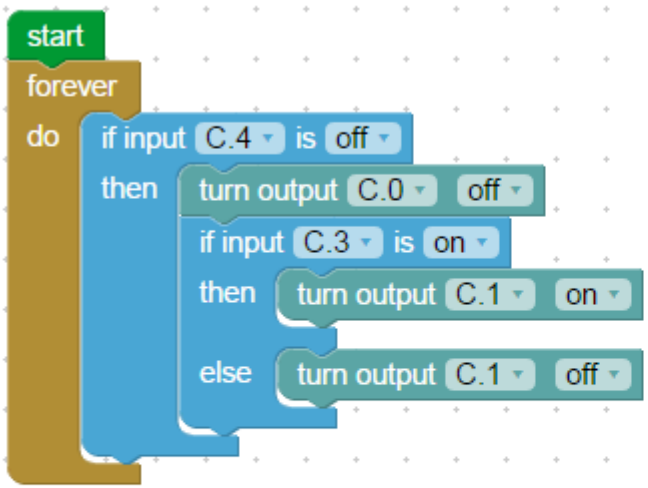
Completed

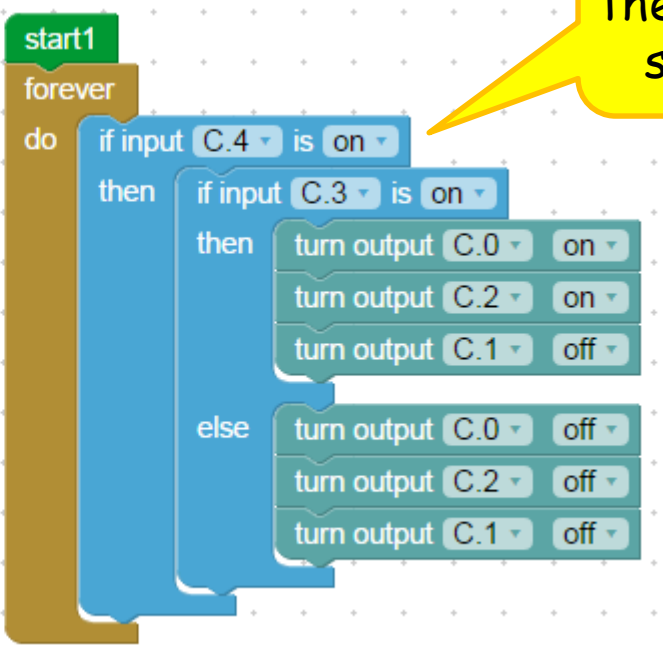
Powertrain: Battery+Solar

Skill needed

CODE THE BATTERY and SOLAR CONTROL (simple)

- Assemble the snap components as shown for "part 1"; and then merge in "park 2" into the circuit design
  - Connect the solar panel (use flex wires if needed)
- Construct the following BLOCKLY code and test it



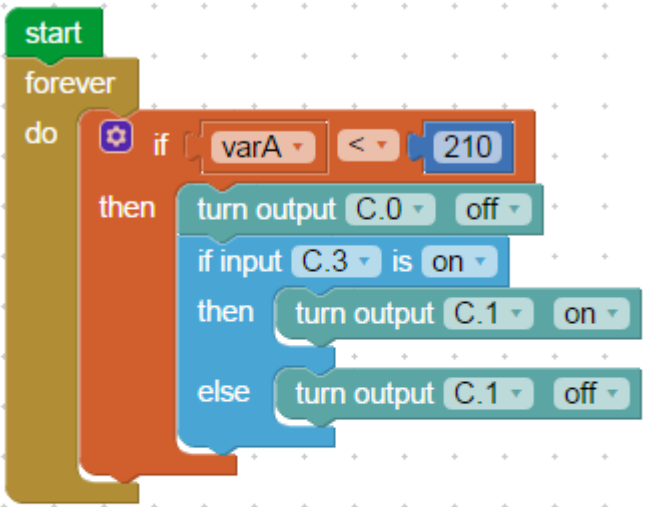


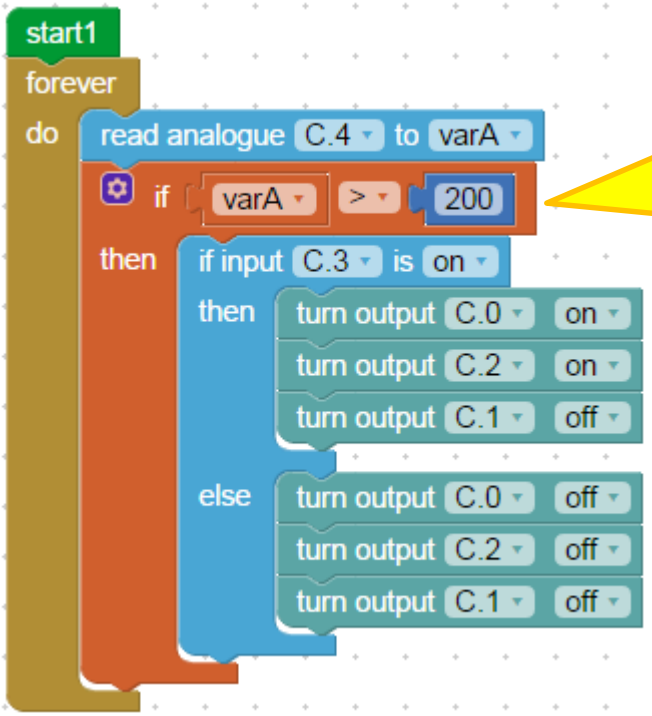
As long as there is some sunlight…

Powertrain: Battery+Solar

CODE THE BATTERY and SOLAR CONTROL (complex)

- Assemble the snap components as shown for "part 1"; and then merge in "park 2" into the circuit design
  - Connect the solar panel (use flex wires if needed)
- Construct the following BLOCKLY code and test it





Analog signal provides more finesse in control of solar drive

Hint: Use DEBUG to see what the sensor value is reading and use that information to adjust the code values

KodeKLIX

www.kodeklix.com

Coding for Young Engineers

Creator:	Nick Coplin
Projects:	www.kodeklix.com/snapcpu4stem/
Microchips:	PICAXE
PICAXE:	www.picaxe.com