Tutorials for Kitronik LAB:bit - Implemented with Insight Mr Bit

K- Tutorial	sonic	mic	pot	button	traffic	dice	speaker	motor	ZIP
switch				(#)					
dice						#			
rainbow			#						#
traffic				#	#				
motor speed			#		#			#	
scary sound		#							
parking sensor	#						#		

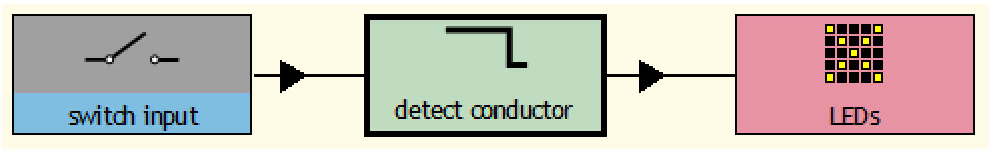
Download HEX files

Download Mr Bit files

1. Switch

Make the micro:bit LEDs show a 'smiley face' when a good electrical conductor is connected to a *switch input*. If a non-conductor or nothing is connected, show a cross.

detect conductor
While the switch input is closed, show the LED image (smile).
While the switch input is open, show the LED image (cross).



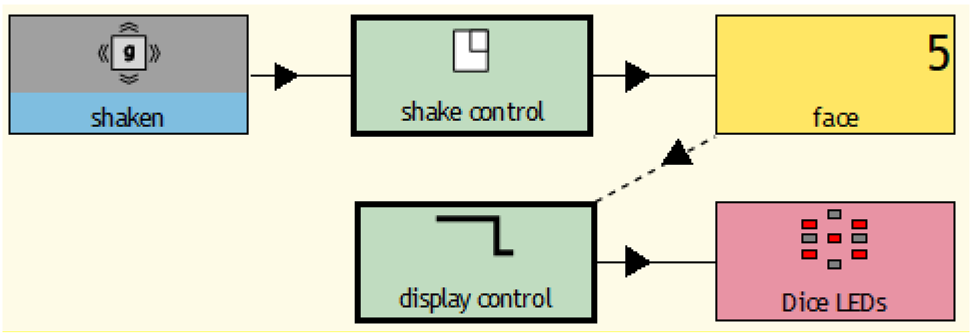
Extension: Use the traffic LEDs to show red for non-conductor and green for conductor.

2. Dice

Make the dice LEDs show one of the six faces of a die at random when the micro:bit is shaken.

show die face
Show the LED pixels array pattern (face) until exit.

shake control
When the micro:bit is shaken, set face to a random number from 1 to 6.

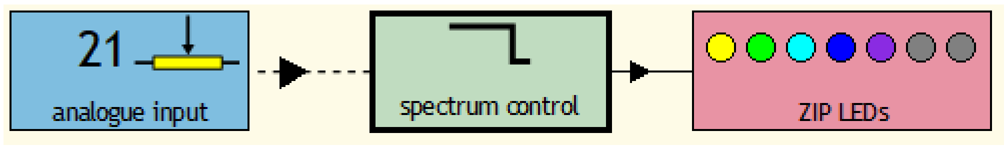


Extension: The micro:bit gesture (shaken) may be replaced by a different gesture, or use a button, or use the sound sensor to detect a hand clap.

3. Rainbow colours

Make the rainbow ZIP LEDs show all seven colours of the spectrum.
Use the potentiometer to vary the rate for the colours to swap places.

spectrum control
Show the LED pixels pattern (rainbow) at analogue input speed until exit.

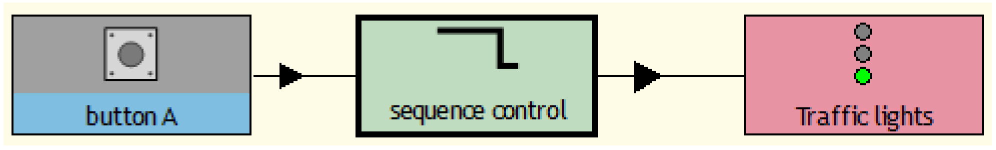


Extension: Experiment with different options for moving the ZIP LED pattern (rotate, shift, scroll etc.).

4. Traffic lights

Starting with the green light, make the traffic LEDs change colour in the correct sequence when a button is pressed.

sequence control
Show the traffic lights pattern (green) until button A gets pressed.
Show the traffic lights pattern (yellow) for 2 seconds.
Show the traffic lights pattern (red) for 8 seconds.
Show the traffic lights pattern (yellow and red) for 2 seconds.



Extensions:
Insert an instruction to delay the change when the button is pressed. You could also show 'WAIT' on the micro:bit LEDs.
Add a second control and second traffic LEDs to make corresponding red and green lights for pedestrians.

5. Motor speed control

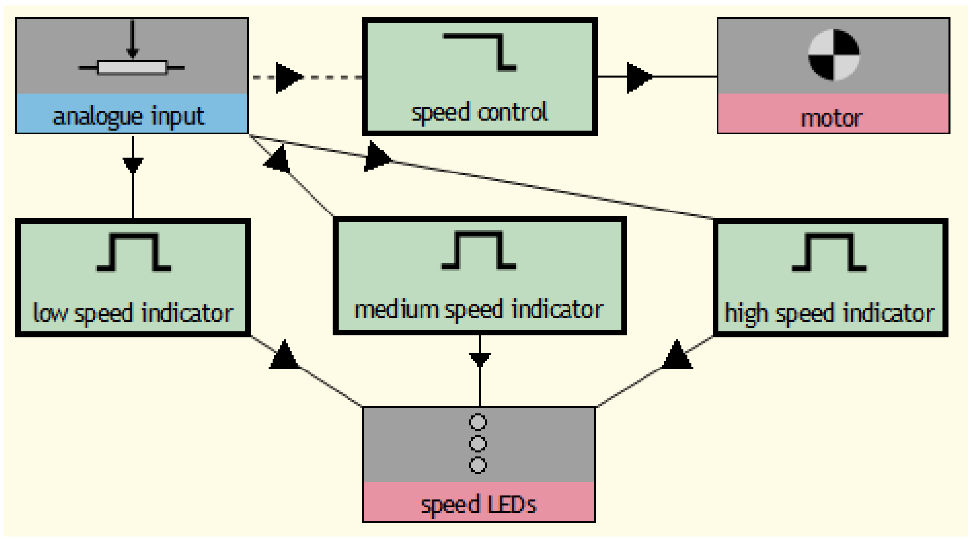
Make the analogue input control the speed of the motor.
Add instructions to make the traffic lights indicate speed: green for low, yellow for medium, red for high.

motor control
While the analogue input is greater than 0, switch on the motor (forwards) (analogue input control).

low speed indicator
While the analogue input is greater than 0 and less than 40, show the speed LEDs pattern (green)

medium speed indicator
While the analogue input is greater than 40 and less than 80, show the speed LEDs pattern (yellow).

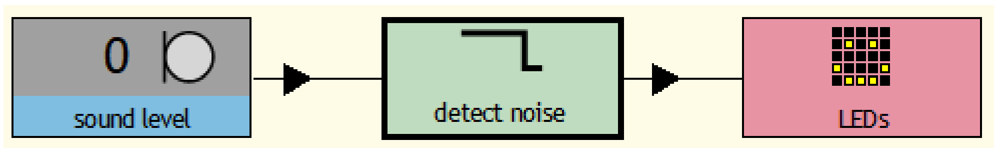
high speed indicator
While the analogue input is greater than 80, show s the speed LEDs pattern (red).



6. Scare the micro:bit

Make the micro:bit LEDs show a ‘happy face’ until the microphone detects a loud noise, when the image changes to a scared face.

detect noise
Show the LED image (happy) until it is noisier than 50.
Show the LED image (scared) for 1 second.

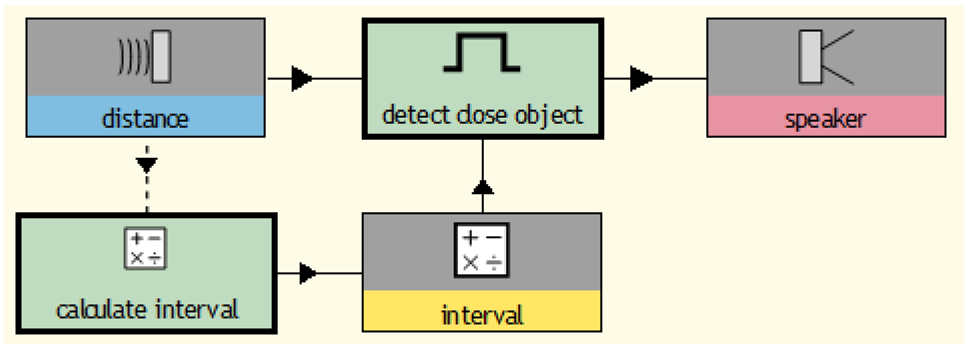


7. Parking sensor

Use the ultrasonic sensor to find the distance of a near object. Make the speaker beep when an object is less than 20cm away. Speed up the beep rate as an object gets closer.

detect close object
When the distance is less than 20, play "beep" on the speaker for 0.3 seconds.
Wait for interval seconds.

calculate interval
Calculate interval = distance / 20 until exit.



Mr Bit Projects for Kitronik LAB:bit

Project	sonic	mic	pot	button	traffic	dice	speaker	motor	ZIP	
rainb-count	#								#	1
rainb- ranger	#								#	2
microwave				#			#	#		3
music box			#				#			4
sound meter		#							#	5
traffic chaos					#					6
rainb-timer				#			#		#	7
Pelican				#	#		#			8
counter-99	#					#				9
double dice				#		#	#			10

Download HEX files

Download Mr Bit files

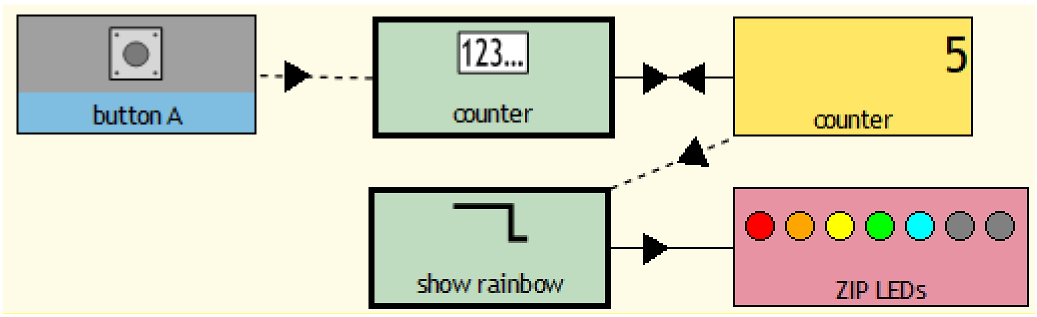
The Mr Bit files show solutions for the projects which can be downloaded to the micro:bit directly. The control modules (green boxes) contain the plain English sentences ('program script').
As a learning strategy for building programs, pupils could first observe the solution code working, then delete the control modules and attempt to recreate them using the editing tools.

1. Rainbow counter

Count how many times a button is pressed and show the result by lighting up the same number of ZIP LEDs.

counter control
Count how many times button A gets pressed until the counter is greater than 7.

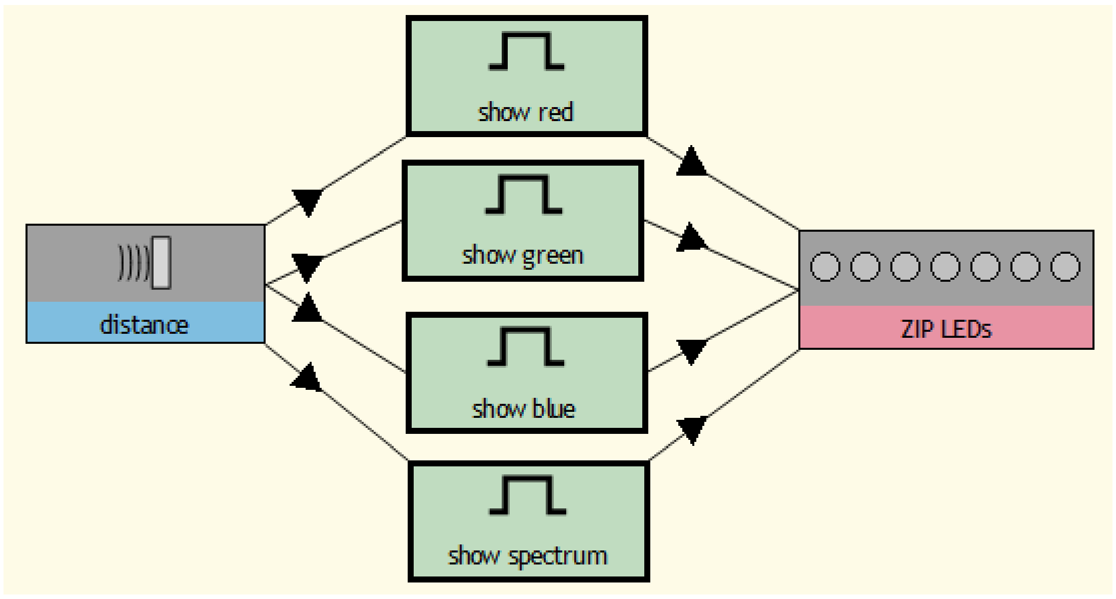
show rainbow
Show the ZIP LEDs array pattern (counter) until exit.



2. Rainbow ranger

The ultrasonic sensor detects the distance of an object that is near the LAB:bit. Use the measurements from the sensor to light up the ZIP LEDs with a different colour according to the distance.

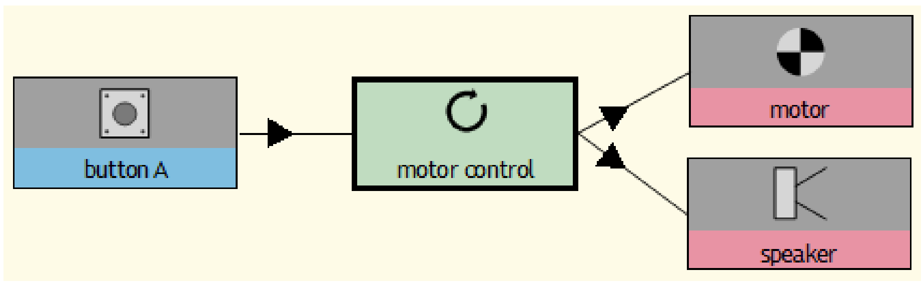
show red
While the distance is greater than 20 and less than 30, show the ZIP LEDs pattern (rotate right r-r-r-r).
show green
While the distance is greater than 30 and less than 40, show the ZIP LEDs pattern (rotate right g-g-g-g).
show blue
While the distance is greater than 40 and less than 50, show the ZIP LEDs pattern (rotate right b-b-b-b).
show spectrum
While the distance is less than 20, show the ZIP LEDs pattern (rotate right roygcbv).



3. Microwave oven

In microwave ovens, washing machines and tumble driers the motor driving the turntable or the washing drum is controlled to rotate alternately forwards and backwards. Design a program to control the motor so that, after pressing button A, its direction changes every few seconds and then sounds bleeps on a speaker when the cycle is finished.

motor control
When button A gets pressed, repeat for 10 seconds:
After 0.5 seconds, switch on the motor (forwards) for 3 seconds.
After 0.5 seconds, switch on the motor (backwards) for 3 seconds.
Repeat once:
Play tones on the speaker for 3 seconds.



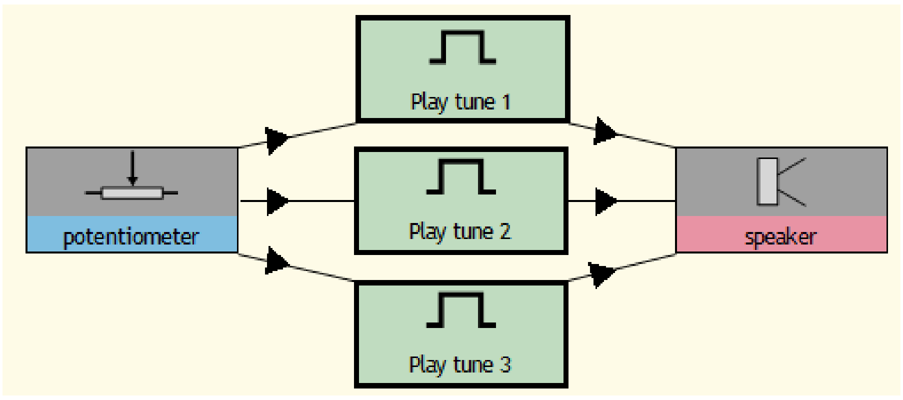
4. Music box

You can use the signal from the potentiometer to select a tune from the speaker:
In Design mode, the music editor shows when you drag a line from the OUTPUTS tab of the module. For each module, choose a tune from the tune library.

Play tune 1
While the potentiometer is greater than 10 and less than 20, play "The Ash Grove" on the speaker.

Play tune 2
While the potentiometer is greater than 20 and less than 30, play "Hornpipe" on the speaker.

Play tune 3
While the potentiometer is greater than 30 and less than 40, play "Happy Birthday" on the speaker.

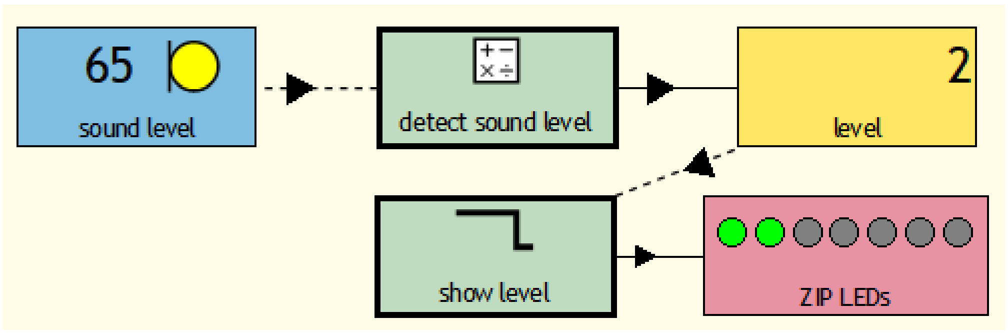


5. Sound meter

Create a program that shows how loud a sound is by lighting up the ZIP LEDs. As a sound gets louder, more LEDs light up. You need to build an array of 7 LED frames. Calculate the array index from the microphone sound level.

show level
Show the ZIP LEDs array pattern (level) until exit.

detect sound level
Calculate level = (sound level - 50) / 7 until exit.

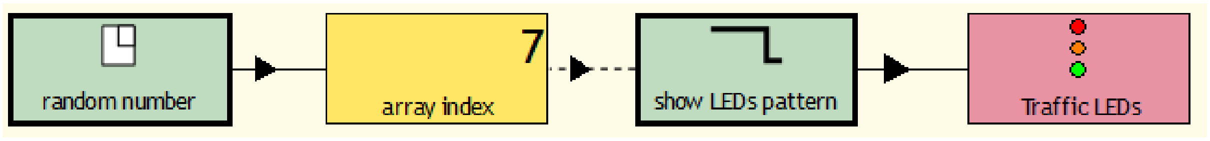


6. Traffic chaos

Normally traffic lights change in an orderly sequence, but you can make them appear in random combinations of one, two and three lights. Build an array of different combinations in 7 LED frames. For the array index, use a variable set to a random number.

random number
After 0.25 seconds, set array index to a random number from 1 to 7.

show LEDs pattern
Show the Traffic LEDs array pattern (array index) until exit.

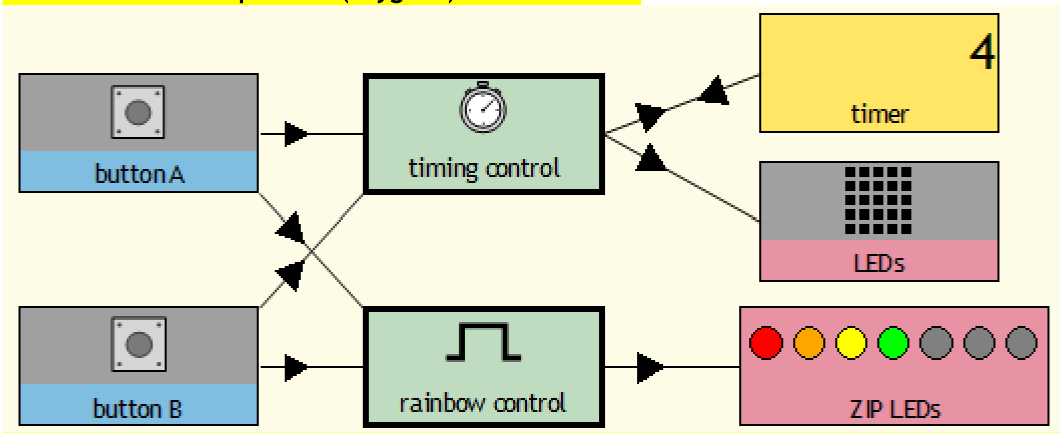


7. Rainbow timer

Use button A to start the Timer module and button B to stop it and show the time on the micro:bit LEDs. Use the same buttons to show a sequence of rainbow colours on the ZIP LEDs while timing is running.

timing control
When button A gets pressed, measure the time until button B gets pressed.
Show the LED number (timer) for 5 seconds.

rainbow control
When button A gets pressed, show the ZIP LEDs sequence until button B gets pressed.
Show the ZIP LEDs pattern (roygcbv) for 5 seconds.

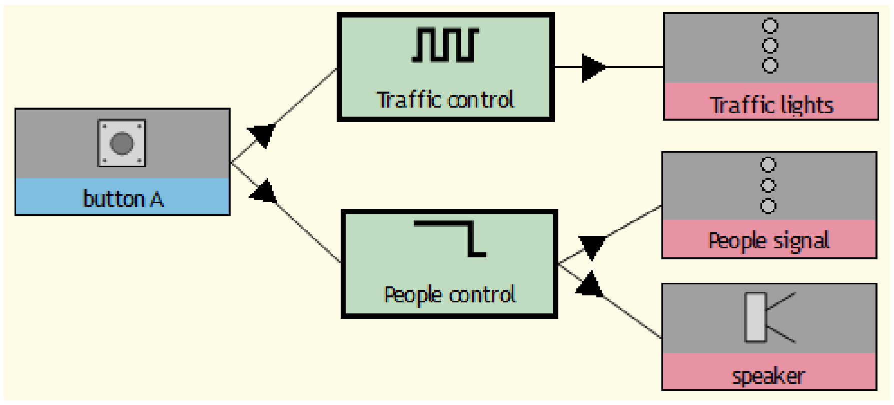


8. Pelican Crossing

Starting with the green light, make one set of traffic LEDs change colour in the correct sequence when a button is pressed. Make the yellow LEDs flash a few times before going back to green.
Use the second set to show the red and green signal for pedestrians. Make the speaker bleep while the green is on.

Traffic control
Show the Traffic lights pattern (green) until button A gets pressed.
Show the Traffic lights pattern (yellow) for 2 seconds.
Show the Traffic lights pattern (red) for 9 seconds.
Flash the Traffic lights pattern (yellow) 3 times.

People control
Show the People signal pattern (red) until button A gets pressed.
Show the People signal pattern (wait) for 2 seconds.
Switch on the People signal pattern (green) and the speaker for 6 seconds.
Flash the People signal pattern (hurry) for 3 seconds.



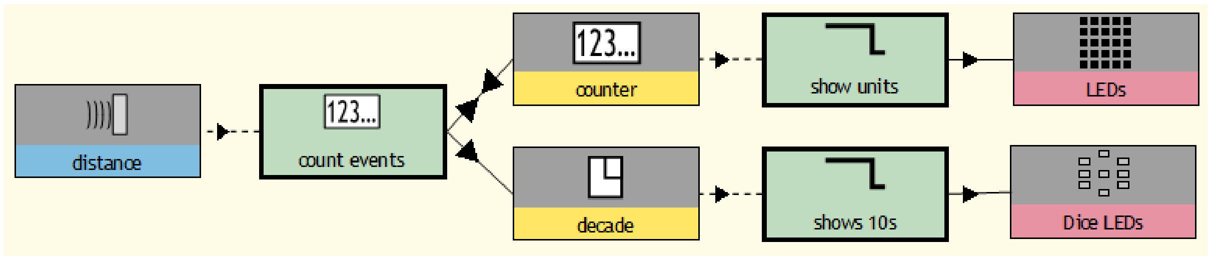
9. Counter-99

Use the ultrasonic sensor to detect people passing by. Count up to 99 people passing close to the sensor (e.g. 30cm) and show the result in tens and units: show the units on the micro:bit LEDs and the tens on the dice LEDs.

count events
Count how many times the distance goes low until the counter is greater than 9.
Add 1 to decade.

show units
Show the LED number (counter) until exit.

shows 10s
Show the Dice LEDs array pattern (decade) until exit.



10. Double dice

Simulate two dice being rolled, the first when button A is pressed and the second when button B is pressed. Every time two numbers are the same, automatically play a tune on the speaker.

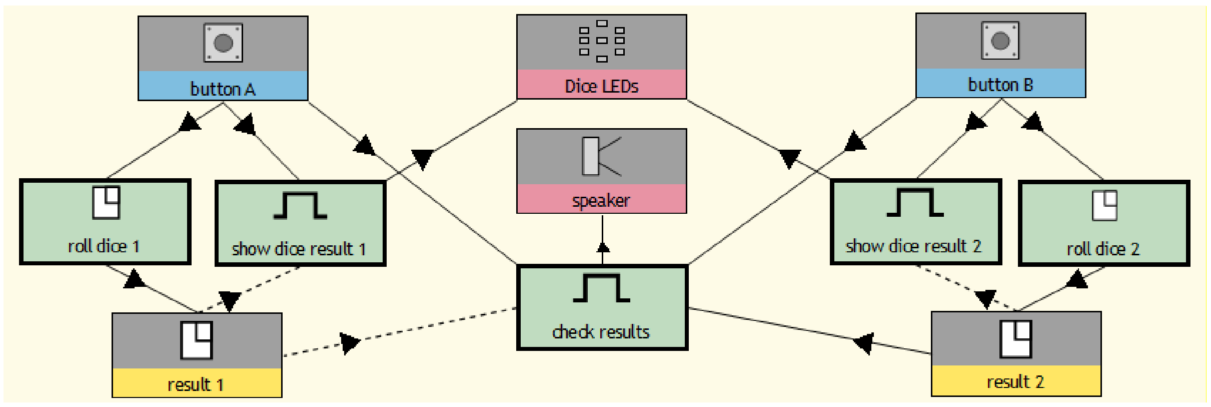
roll dice 1
When button A gets pressed, set result 1 to a random number from 1 to 6.

show dice result 1
When button A gets released, show the Dice LEDs array pattern (result 1) for 2 seconds.

roll dice 2
When button B gets pressed, set result 2 to a random number from 1 to 6.

show dice result 2
When button B gets released, show the Dice LEDs array pattern (result 2) for 2 seconds.

check results
Wait until button B gets pressed or button A gets pressed.
When the result 2 is equal to (result 1), play tones on the speaker for 5.625 seconds.



APPENDIX – Topics for starter tutorials videos

1 Selecting sensors and devices from Gallery in Scene view
1 System and Scene views
1 Modes [Design, Run, Control]
Choosing micro:bit - firmware, LAB:bit interface board.
1 Basic input-output system [Design mode, tabs, links, conditions]
8 Counter module
4 Dialogue for Traffic LEDs
2 Dialogues for Dice LEDs
3 Dialogues for ZIP LEDs
5 Analogue sensors in Run mode, slider adjuster, threshold, start values
6 Sound sensor
2 Creating variables
2 Random number variables
5 Variables controlling LEDs, animations and motor speed
9 Showing numbers/images on LEDs - dialogue - display types
7 Music editor with speaker - show how to get a beep
Repeat module
4 Creating a sequence
Pulse module for flashing LEDs
8 ZIP LEDs array - frames and index
2 Dice LEDs array
9 Timer module
7 Calculation module

Videos

Tutorial 1:	https://youtu.be/3OMRCZJM5pc	Inputs and outputs
Tutorial 2:	https://youtu.be/xndCctCDCdQ	Random numbers and image array
Tutorial 3:	https://youtu.be/WyJ5bMPAGfg	Controlling ZIP LED colours
Tutorial 4:	https://youtu.be/E7jYcPik43Y	Instruction sequences
Tutorial 5:	https://youtu.be/5pPr_XRS1cE	Analogue control
Tutorial 6:	https://youtu.be/5BDvlVw3M44	Sound and images
Tutorial 7:	https://youtu.be/6v39f_EWGcs	Ultrasonic distance sensor
Tutorial 8:	https://youtu.be/b_2_o4Es9MY	Counting events
Tutorial 9:	https://youtu.be/S2fSHqWGPF0	Measuring time
Projects:	https://youtu.be/fQBEHESlxQc	