

Activity 6

Student Worksheets

CHALLENGES FOR TODAY

Today you are going to explore how the yellow sensor blocks are used in conjunction with the Maths Block. You will also use the Loop Block.

CHALLENGE 1

Over the course of three challenges, you will be programming your wheeled robot to simulate a car's parking sensor.

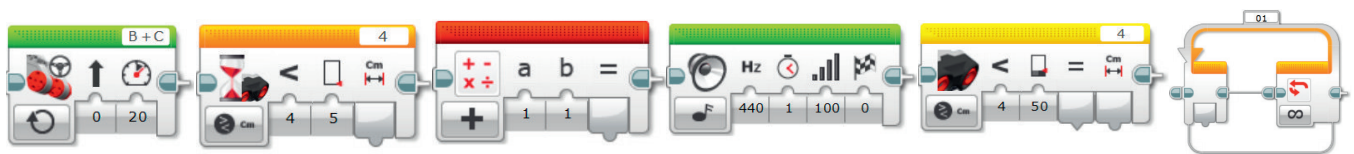
What happens when some cars reverse? There is a beeping sound, which becomes quicker as the car gets closer to an obstacle.

Can you devise a program that drives your wheeled robot backwards, emits beeping noises as it approaches an obstacle and then stops automatically at a set distance away from the object?

Tip 1: You will need to use parallel programming (multitasking).

Tip 2: You will need to use what you have learnt about the Maths Block and Data Wires, in order to increase the frequency of the beeps as your wheeled robot gets closer to the obstacle.

Blocks to Consider



Plan your program first. Write it in pseudo-code below:

Activity 6

Student Worksheets

CHALLENGE 2

What have you noticed about your program and in particular the beeping sounds?

They should become quicker as your wheeled robot approaches the obstacle.

However, in real life, the warning sounds only begin when the vehicle is a certain distance from an obstacle.

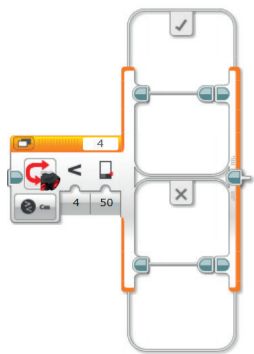
Can you simulate this in your program?

You will need to build on the program you have already created, but alter it slightly so that the beeping begins at a given distance from the obstacle.

Tip: You will need to utilise a true/false statement and Boolean logic. Which programming block do you need to use for this?

Blocks to Consider

Use the same blocks that you used in programming task 1, but also consider using the following:



Plan your program first. Write it in pseudo-code below:

Activity 6

Student Worksheets

CHALLENGE 3

By now your wheeled robot should be simulating rear parking sensors more and more.

Now it's time to take your programming one step further.

You will need to add two more features:

1. Can you make the beeping sound stop when your wheeled robot stops at a given distance away from the obstacle?
2. Can you make your wheeled robot slow down as the beeping sound starts?

Tip 1: In order for the beeping sound to stop, you will need to interrupt the loop.

Tip 2: You will need to map speed to distance by using a second Maths Block somewhere in your program. Can you work out where?

Blocks to Consider

Use the same blocks that you used in programming tasks 1 and 2, but also consider using the following:



Plan your program first. Write it in pseudo-code below:

Activity 6

Student Worksheets

After a programming activity, it is important to note down your thoughts and observations. Consider the following points and then in the box below record how the activity went.

- How could you improve your program?
- Could your program have been more streamlined? Have you used too many blocks? Is there a more efficient way of building your program?
- What examples of real-world applications could you see your program being used in?

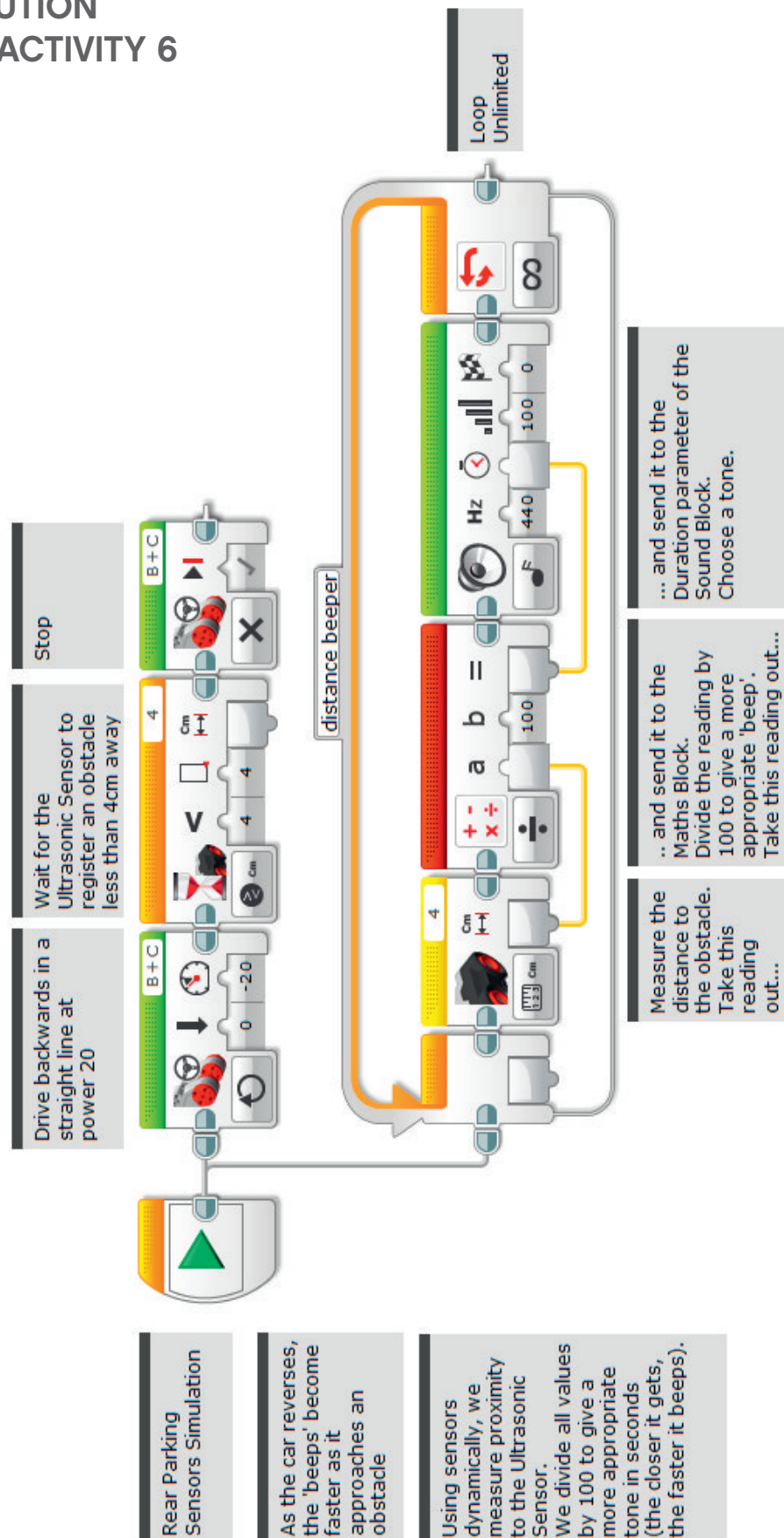


Thoughts and Observations

Activity 6

Appendix

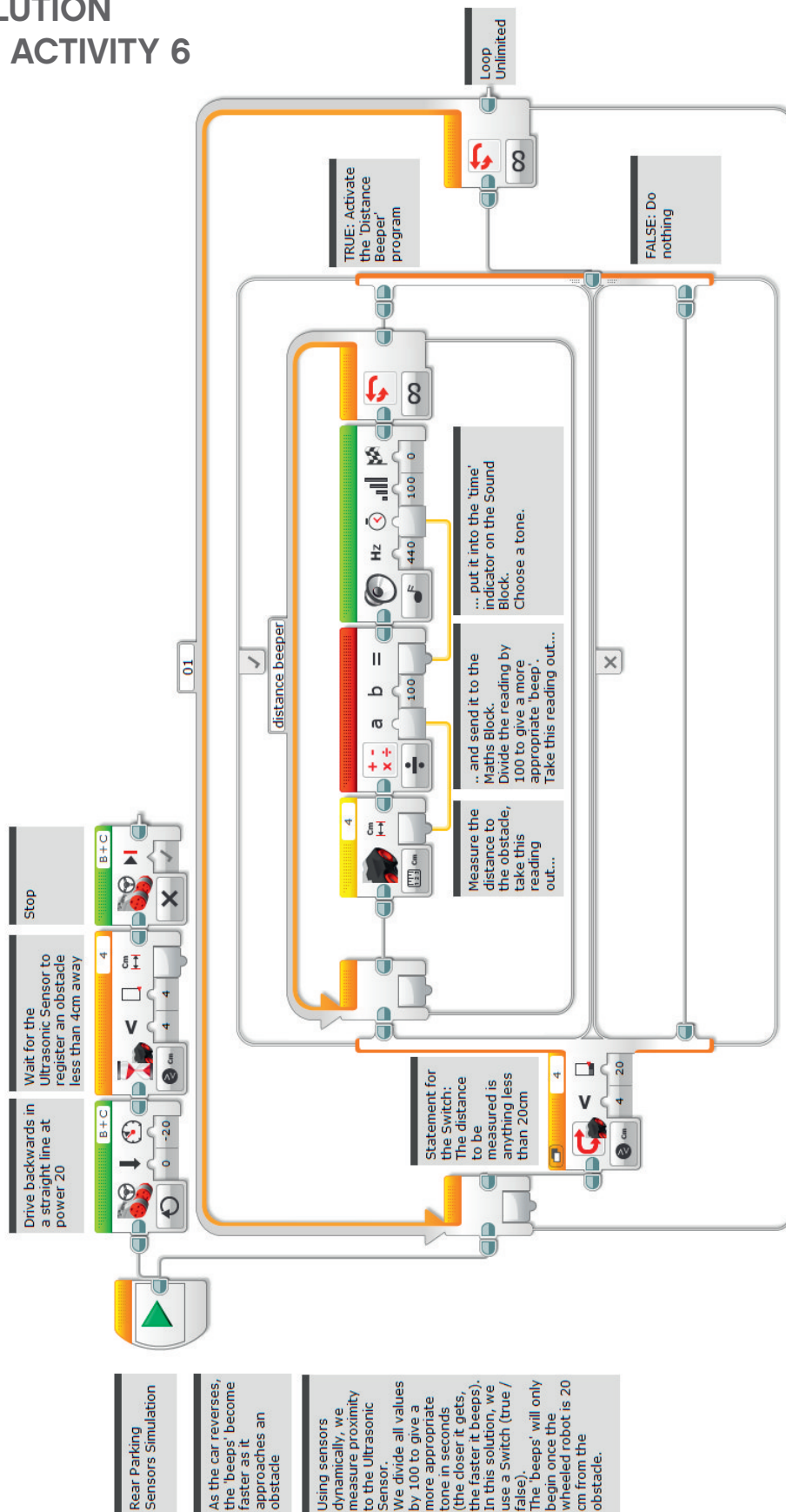
POSSIBLE SOLUTION
FILENAME: CS ACTIVITY 6
TAB: MAIN 1



Activity 6

Appendix

POSSIBLE SOLUTION
FILENAME: CS ACTIVITY 6
TAB: MAIN 2



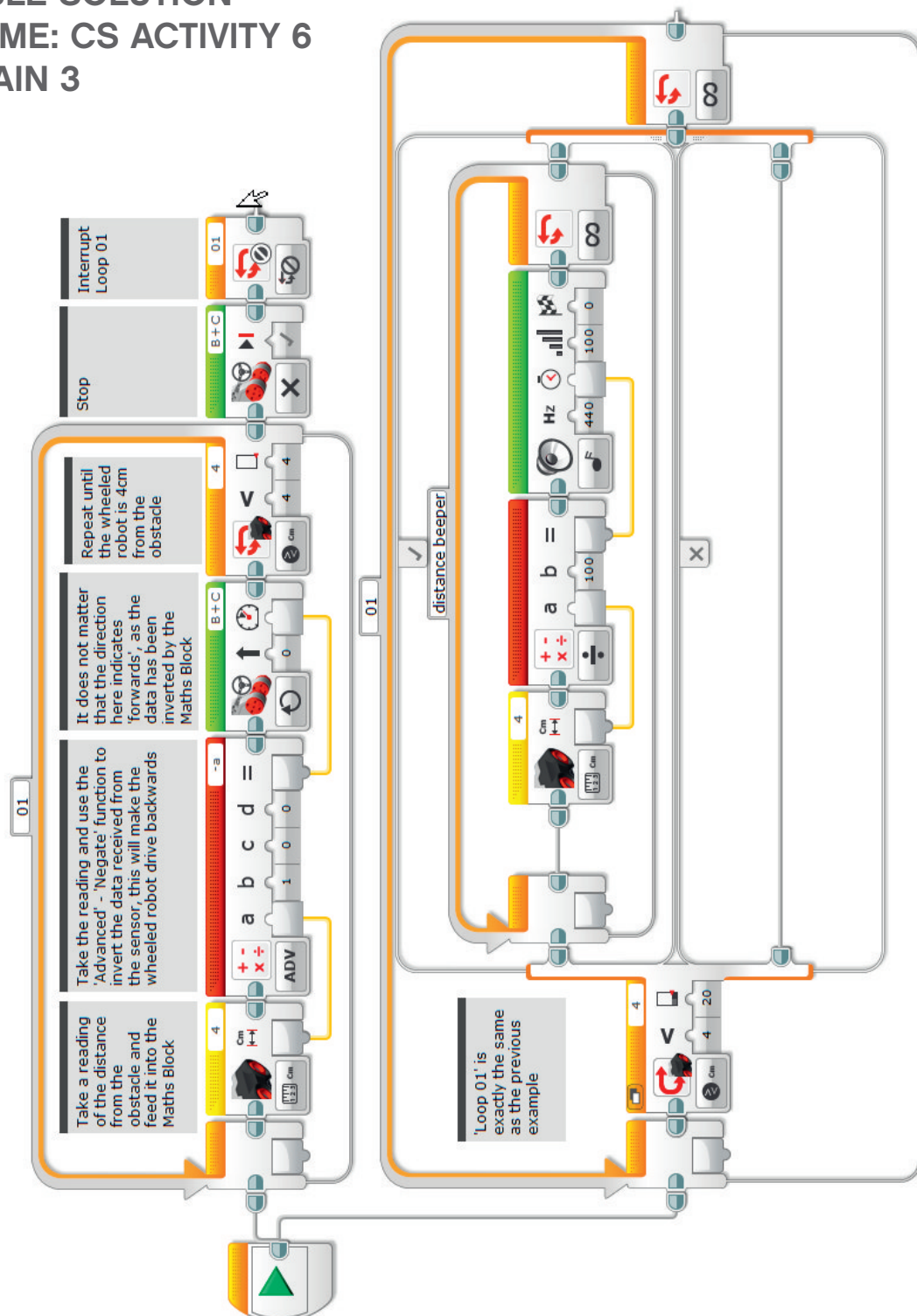
Activity 6

Appendix

POSSIBLE SOLUTION

FILENAME: CS ACTIVITY 6

TAB: MAIN 3



Rear Parking Sensors Simulation

As the car reverses, the 'beeps' become faster as it approaches an obstacle

Using sensors dynamically, we measure proximity to the Ultrasonic Sensor. We divide all values by 100 to give a more appropriate tone in seconds (the closer it gets, the faster it beeps). In this solution, we use a switch (true / false).

The 'beeps' will only begin once the wheeled robot is 20 cm from the obstacle. This time, the wheeled robot will begin to slow down as it approaches the obstacle. When it stops (at 4cm from the obstacle) the 'beeping' will also stop.

Activity 6

Appendix

Possible ROBOTC Solution

FILENAME: Activity6_1.c

```
Activity6_1.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard      !**//
3
4  /*
5  Create a program to drive the robot backwards with a warning beep getting quicker
6  as the Ultrasonic Sensor sees an approaching object. The robot stops when a distance
7  of less than 4cm is reached.
8  */
9
10 task distanceBeeper()
11 {
12     while(true)
13     {
14         //Start playing a tone (in tens of milliseconds).
15         playTone(440, getUSDistance(sonarSensor));
16
17         // Delay while the sound plays.
18         while(bSoundActive)
19         {
20             sleep(10);
21         }
22
23         //Add 50ms of "silence" to avoid a constant tone.
24         sleep(50);
25     }
26 }
27
28 task main()
29 {
30     //Start the "distance beeper" task.
31     startTask(distanceBeeper);
32
33     // Set the motor speed.
34     setMotorSpeed(motorB, -20);
35     setMotorSpeed(motorC, -20);
36
37     // Wait while the distance is great than 4.
38     while(getUSDistance(sonarSensor) >= 4)
39     {
40         sleep(10);
41     }
42 }
43
```

This code may be subject to change as ROBOTC is updated periodically.

Activity 6

Appendix

Possible ROBOTC Solution

FILENAME: Activity6_2.c

```
Activity6_2.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard    !!*/
3
4  /*
5   Create a program to drive the robot backwards with a warning beep getting quicker
6   as the Ultrasonic Sensor sees an approaching object. The beep only sounds at a distance
7   less than 20cm. The robot stops when a distance of less than 4cm is reached.
8   */
9
10 task distanceBeeper()
11 {
12     while(true)
13     {
14         if (getUSDistance(sonarSensor) < 20){
15             //Start playing a tone (length determined by distance).
16             playTone(440, (getUSDistance(sonarSensor)));
17
18             // Wait for the tone to be done playing.
19             while(bSoundActive)
20             {
21                 sleep(10);
22             }
23
24             //Add 50ms of "silence" to avoid a constant tone.
25             sleep(50);
26         }
27     }
28 }
29
30 task main()
31 {
32     //Start the "distance beeper" task.
33     startTask(distanceBeeper);
34
35     setMotorSpeed(motorB, -20);
36     setMotorSpeed(motorC, -20);
37
38     while(getUSDistance(sonarSensor) >= 4)
39     {
40         //Keep driving.
41         sleep(10);
42     }
43 }
44
```

This code may be subject to change as ROBOTC is updated periodically.

Activity 6

Appendix

Possible ROBOTC Solution

FILENAME: Activity6_3.c

```
Activity6_3.c
1  #pragma config(StandardModel, "EV3_REMBOT")
2  /**!!Code automatically generated by 'ROBOTC' configuration wizard  !!*/
3
4  /*
5   Create a program to drive the robot backwards with a warning beep
6   getting quicker as the Ultrasonic Sensor sees an approaching object.
7   The beep only sounds at a distance less than 20cm. The robot
8   slows down as it nears the object. The robot stops when a distance
9   of less than 4cm is reached as does the beeper.
10  */
11
12  task distanceBeeper()
13  {
14      while(true)
15      {
16          if (getUSDistance(sonarSensor) < 20){
17
18              //Start playing a tone (length determined by distance).
19              playTone(440, (getUSDistance(sonarSensor)));
20
21              //Delay for the length of the tone.
22              //Wait while the sound is playing.
23              while (bSoundActive)
24              {
25                  sleep(10);
26              }
27
28              //Add 50ms of "silence" to avoid a constant tone.
29              sleep(50);
30          }
31      }
32  }
33
34  task main()
35  {
36      //Start the "distance beeper" task.
37      startTask(distanceBeeper);
38
39      while(SensorValue(sonarSensor) >= 4)
40      {
41          setMotorSpeed(motorB, -getUSDistance(sonarSensor));
42          setMotorSpeed(motorC, -getUSDistance(sonarSensor));
43          sleep(50);
44      }
45
46      //Set motor speed to 0 and stop the distanceBeeper loop.
47      setMotorSpeed(motorB, 0);
48      setMotorSpeed(motorC, 0);
49      stopTask(distanceBeeper);
50  }
51
```

This code may be subject to change as ROBOTC is updated periodically.