## Lesson 20 – Activity Sheet

## Getting Started

## An ultrasonic sensor works by sending out an ultrasonic sound wave as a ***Ping*** and wait to see how long it takes for the sound wave to reflect off a solid object and be echoed back. This time can then be converted into distance

* We can use the **Sonar** extension in MakeCode to enable us to use the ultrasonic sensor
* There is also a pre-built block in the Bit:Bot extension
* If you are using the sensor without the Bit:Bot chassis you need to ensure you have 5v powering the sensor – the standard output from the micro:bit is only 3v and might not be enough to power the sensor reliably
* It is not just pedestrians and other road users we need to avoid in our self-driving car but also road & street furniture. Road and street furniture is the term used for objects and pieces of equipment installed along streets and roads for various purposes
* Road furniture includes benches, traffic barriers, bollards, post boxes, phone boxes, streetlamps, traffic lights, road signs, bus stops, war memorials and waste receptacles
* The design and placement of furniture takes into account aesthetics, visual identity, function, pedestrian mobility and road safety

## Success Criteria

## Understand how an ultrasonic sensor can be used to create a collision avoidance system

## Calculate the distance to an object

## Design an algorithm for a collision avoidance system

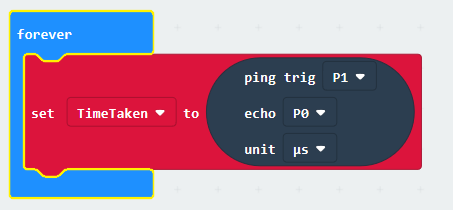
## Create and test an algorithm for a collision avoidance system

|  |  |  |
| --- | --- | --- |
| **Ping Time** | **Calculation** | **Distance in cm** |
| 0.05 Second |  |  |
| 2.3 Seconds |  |  |
| 0.5 ms |  |  |
| 1.5 ms |  |  |

Order the algortihm so it correctly stops the vehicle 10 cm before the wall

|  |  |
| --- | --- |
| **Pseudocode** | **Pseudocode** |
| If Distance to Object < 10 cm then |  |
| Transmit Ping Signal |  |
| Start timer |  |
| Motors Off |  |
| Calculate Distance that sound travelled |  |
| Stop timer |  |
| End |  |
| Calculate Distance to Object |  |
| Start |  |
| Else |  |
| Calculate time taken |  |
| Receive Ping Signal |  |
| Motors on |  |

* Use your algorithm and the code snippet below to test your algorithm functions correctly by getting your Bit:Bot to stop 10 cm from the classroom wall
* This is measuring in microseconds so you will need to adjust your formula accordingly



## Pro-tip

* When completing **pseudocode**, it makes it easier to read if you tab in your selection and iteration statements similar to how you might in Python
* Experiment with the **Sonar** block – can you make it do the calculation for you?

## Test Time

* What happens if the object you are avoiding is less than 3 cm in height?
* What happens is the object you are avoiding is hollow?
* What happens if the object you are avoiding is soft?

## Stretch Tasks

* How will vehicle speed affect your distance calculations?
* What is the shortest distance you can measure without crashing into a wall
* What is the fastest you can travel to stop 2 cm from the wall?

## Final Thoughts

* Today we have looked in a little more detail how the ultrasonic sensor works
* We have looked at how we calculate distance and produced an algorithm to make sure a robot stops before hitting an object
* We have experimented using the ultrasonic sensor and micro:bit and tested our algorithm