

Analog sensor

Reading at 1m, 50cm 30cm 20cm 15cm 10cm 5cm

Team work, somebody hold the sheet of paper and others record the values. Take turns

Notice that the largest values are taken at around 15cm

smallest values at 1m

Values get smaller shorter than 15cm

This is a limitation of the sensor we are using. Therefore we must find a way to protect against this limitation becoming a hinderance or a problem in the product we envisage.

Remember we are looking to create a robot that will drive around randomly but not crash into things. No matter what sensor we use there will be some limitations. The pros of this sensor are with the reading of values from it. It is very easy. The cons are the accuracy. The datasheet shows exactly how the distance an object is relates to the voltage that the sensor provides.

When preparing our robot we must ensure that it doesn't mix up 8cm and 30cm, because if it thinks 8cm is 30cm it will stay driving on and then crash when it thinks the wall is far away.

So if we say we will stop the robot 30cm from the wall, we want it to watch for readings greater than the value recorded at 30cm.

So we create a threshold value that becomes the value that we don't want the readings to get greater than. If the values get greater than that we must stop the car.

We now will do things slightly backwards. It is typical of an engineering situation, that you are working on one part of the project, and you need to do something, like stop the car. But you don't know how to yet. Nor do you know how to drive the car. But you don't need to know how the gears and engine all work to actually drive a car, you only need to know where the clutch and accelerator are. Similarly here, we don't know how we are going to stop the car yet, but we can still just say something that might stop the car. So we will call a function, like digitalWrite, and it will do the stopping for us. Stop().

But stop() doesn't exist, so we have to make it.

Now we will learn how to make the car move.

So if we have a stop function we must also have a move function, so we will make forward and reverse functions also.

how PWM works

How we program for PWM (max200)

How we program the motors

Now up to you to program the motors to go backward and stop

Cables

Be very careful!!! Ground and Voltage must not touch. The wire is quite long so will stick out of the terminal, make sure they are done tidily. Get the polarity correct!

Plug out the computer cable from the arduino.

Plug in the battery and lets see our robots moving. Check that the stop by putting a block in front of the sensor, watch how they don't stop if the block is too close.

Last bits, turning, using reverse. and off they go. Watch chair legs, they don't see them.