0.1 AVOIDING OBSTACLES

0.1.1 Tasks:

Write the program to drive the robot around the class and avoid the obstacles.

1. Check the value of distance sensor. If the distance is greater than ...
2. ... the robot can drive forward.
3. ...else ... the robot must to stop/go back/turn.

Program 1: Avoiding Obstacles.

```
#include "RobotMovingFunctions.h"
const int DIST_SEN_PIN   = A0;
const int DISTANCE_LIMIT = 20;
void setup()
{
  setIOpins();
  pinMode(DIST_SEN_PIN,INPUT);
}
float getDistance_cm()
{
  int adc_value  = analogRead(DIST_SEN_PIN);
  float distance = 1/(0.045 * 5.0/1024 * adc_value);
  return distance;
}
void loop()
{
  if ( getDistance_cm() > DISTANCE_LIMIT )
  {
    moveForward();
  }
  else
  {
    stopTheRobot();
  }
}
```

0.1.2 Questions:

1. What are the values of the distance sensor (use `Serial.println(distance)` to verify)?
2. Robot stil hits the obstacles that are not in view angle of the distance sensor. Write and use new function for moving the robot forward more carefully.

0.1.3 Summary:

0.1.3.1 Moving the robot and checking the sensor simultaneously The main important proces in robotics is S-R-A loop. This process is used in different situations and many times. One can be where we are moving the robot forward and at the same time observing the sensors value with the intention to stop it when the specific condition is met.

```
   void goForwardCarefully()
   {
      for (int i = 0; i < 10; i++)
      {
         robotLeft();delay(50);
         if (getDistance_cm() < DISTANCE_LIMIT) brake;
      }

      for (int i = 0; i < 10; i++)
      {
         robotRight();delay(50);
         if (getDistance_cm() < DISTANCE_LIMIT) brake;
      }
   }
```

<++>

0.1.4 Issues:

0.1.4.1 <++> <++>