Create a new robot controller with all of the following functions:

* 1. void getShortDist();
  2. void getDist();
  3. void getColor();
  4. double getLHeat();
  5. double getRHeat();
  6. double getAngle();
  7. void getGPS();
  8. void setMotors(double leftSpeed, double rightSpeed);
  9. void goForward(double dist);
  10. void goBackward(double dist);
  11. void turnR();
  12. void turnL();

All sensor and motor variables should be global

getShortDist is for the 8 distance sensors on the robot already (they have very short range). getDist is for 4 regular distance sensors you will have to add on.

getShortDist, getDist, getColor, and getGPS should modify global arrays storing the appropriate values

Like before, you can combine goForward and goBackward and just allow a positive or negative input for distance

You can also do the same with turnR and turnL – just have a parameter so you can pass in what direction you want to turn

turnR and turnL should both be 90 degree turns using the gyro, not encoders

When you are done: implement a wall tracing program into the maze world. It is up to you as to how to deal with the black pits.

After you finish your own structure:

Create a typedef struct called Tile and give it the following attributes:

* N – 1 bit
* E – 1 bit
* S – 1 bit
* W – 1 bit
* vis – 1 bit

We will explain the meaning behind this in a little bit

Create an array of Tiles called maze or map or whatever you want

Create a getSurroundings function that takes in the Section you are in and Orientation you are facing, and fills out the maze array accordingly.

void getSurroundings(int Section, int Orientation){

//1. Probes long range distance sensors.

//2. Fills out maze array accordingly

//Make sure you mark the correct bits in the correct element of the array

//Section 4 should match with Maze[4]

}

Next step:

Create a ifWall function that takes in a Section and Orientation and checks if there is a wall in that section in that orientation.

Bool ifWall(int section, int orientation) {

//If section is 5 and orientation is 2, returns true if there is a wall to the south in section 5

//returns false otherwise

//Uses Maze array to check

}

When finished:  
Polish your turn functions so that the function can be called many times in a row and still work (you cannot just wait until the gyro is larger than 90)

Work on your wall tracing program and try and think of a way to deal with the black pit