IDP M107 SOFTWARE DOCUMENTATION

Code structure and algorithms - Link to final code listing

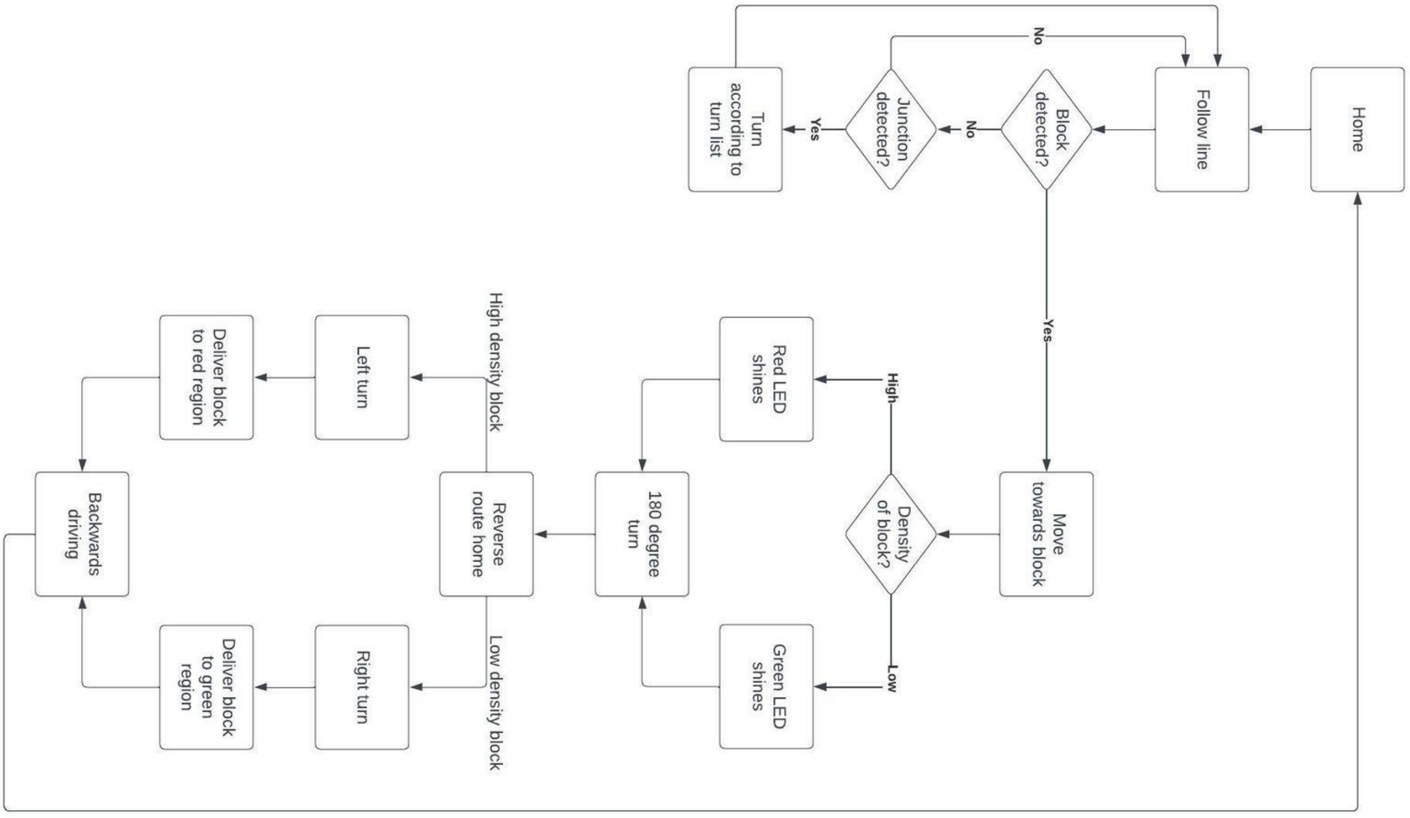
(alternate link: https://github.com/Ben-Fenocchi/IDP-software)

Function Name + arguments	Purpose	Location called
speedLeft()	Controlling left motor	
speedRight()	Controlling right motor	
turn180()	After retrieving the block, we can use the same path to avoid coming across another on the return journey	After retrieval of a block
rightTurn()	90° turn for line following purposes	Inside of moveToJunction()
leftTurn()	90° turn for line following purposes	Inside of moveToJunction()
followLine()	Adjust chassis to move along the line	Inside of moveToJunction()
moveToJunction()	Moves to the next junction in our preset path	In main loop
checkForBlockDownwards()	Monitors reading from tof sensor to detect block	In main loop
blockBurglar()	Close the “hugger” over the block so it can be safely transported	Inside of checkForBlockDownwards()
ETgoHome()	Once collected a block, this function returns the robot to the home square	Inside of blockBurglar()
alignChassis()	Aligns the chassis with the axles parallel to a line, useful for travelling from the starting box to the green/red ones	Inside of ETgoHome()
deliverBlock()	Moves from start to correct destination, deposits block, and resets position for the next run	Inside of ETgoHome()

We have used a modular approach to our software.

By writing each main process as its own subroutine, we ensure modularity and efficient debugging.

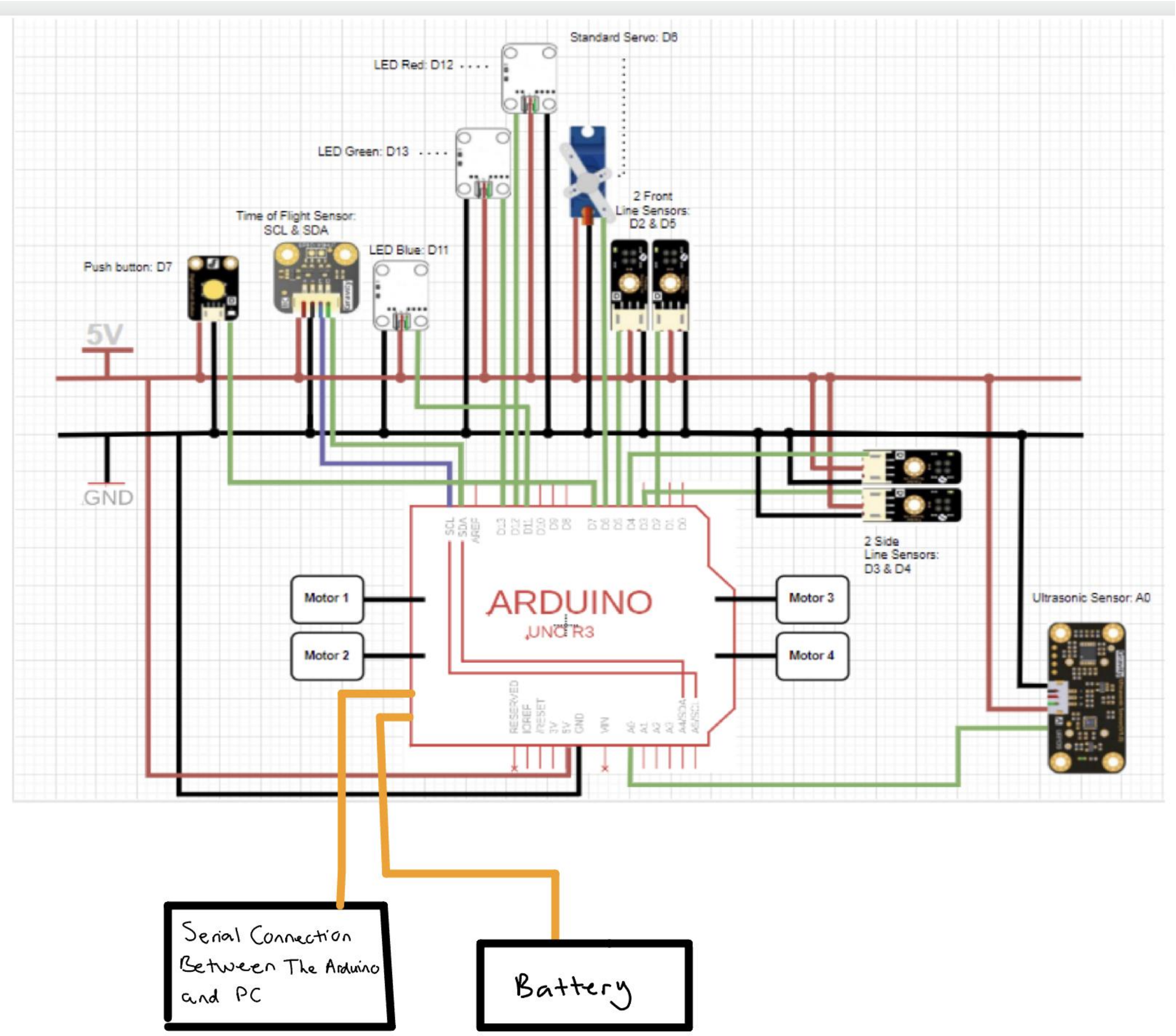
It also increases code maintainability and scalability.



IDP M107 SOFTWARE DOCUMENTATION

Code structure and algorithms - Link to final code listing

System diagram:



The green LED will illuminate for a higher density block, and red will for a lower density. We do this by writing HIGH/LOW to the respective pins after block detection to signify i)finding a block ii) the density.