Function Psudo Code for ‘four in a line’ sensor placement

N.b all turning done is differential (tank-like) and turning right is positive (d>0)  
Red describes how control alg decides to perform this action

Line-Following:

* Speed of left motor = x + d
* Speed of right motor = x – d
* Where x is the base velocity either set to a constant slow forward role, or could be proportional to rate of feedback (so that robot speeds up on straight sections and slows down on turny bits)
* d, the differential is our negative feedback variable and will be -Cv\_left+Cv\_right where v\_left is voltage in left line sensor and v\_right is voltage measured in right line sensor

90o Turn:

* Back sensor and far right (far left) sensor are activated (digitally?)
* Line following deactivated
* x => 0, d => constant (-constant)
* …until far right (far left) sensor turns on
* Rejoin line algorithm (E.g assume slight overshoot, let d = small positive constant (-ve in case of left turn) and perform line rejoin algorithm

180o Turn: (potentially do 90o turn then repeat for however long that took)

* After block picked up or placed down
* Line following deactivated
* x => 0, d => constant
* …until back sensor turns on (digitial)
* Perform line rejoin algorithm
* (Alternatively if robot started with a reading v\_left and v\_right on left line sensor and right line sensor respectively, could just turn until readings are reversed(currently not sure how to reliably rejoin lines after turns…)possibly can be combined with above method)

Tunnel-Junction(1st time):

* After blue block picked up
* And T-junction navigated
* Favour right line sensor when both sensors trigger
* Just a slight modification to line following that says when both line sensors are on, ignore left line sensor reading
* (Important to turn this modifier off before Blue T-junction)
* (Potentially could use this modifier at beginning of course to reduce chance of treating tunnel-junction as T-junction)

Detecting Target Zones without block:

* If far left and far right triggers (as with T-junction) having recently performed a T-junction (i.e just been line following since) then must be a target spot
* Line following deactivated
* Set d to 0, travel forward for a bit and then perform line joining algorithm

Detecting Target Zones with block:

* If far left and far right triggers (as with T-junction) while still on pill, then must be a target spot.
* Line following deactivated
* (Assuming block is a known distance in front of far line sensors)
* Reverse this set distance
* Release claw
* Reverse some more
* Perform 180o turn
* (note that depending on dimensions of track, especially with blue target zones, this might be tight/not work)

Line rejoin algorithm assuming we know which side the line is on:

* Automatically if both line sensors reading 0 for period T or if turned on manually after manouvre
* Set d to small constant that steadily increases (i.e robot falls towards line)
* Set x to some constant – not too small not too big (maybe also steadily increases)
* First line sensor to register will be rightside line sensor if d > 0 and line on right-hand side (or left if d< 0 and line of left-hand side of robot)
* Wait