Left Override	Right Override	Left Eye	Right Eye	Left Motor	Right Motor
0	0	0	0	1	1
0	0	0	1	0	1
0	0	1	0	1	0
0	0	1	1	1	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	1	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	1	0
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

- Lo = Left Override
- Ro = Right Override
- L = Left Eye
- R = Right Eye
- Lm = Left Motor
- Rm = Right Motor

Left Motor (Simplified) Boolean Expression

- Lo + L + R`.Ro`

Right Motor (Simplified) Boolean Expression

- R + Ro + L`.Lo`

The overrides are used to divert the buggy when it reaches a fork (i.e. at the pit lane), if the override is on the buggy will turn in the opposite direction of the named override (e.g. will turn right when left override is on). This means that the opposite motor only enables when it sees white (Logic 1), to stop the buggy turning too much.

I feel the override may need to be triggered when both eyes see black (Logic 0), as this should only occur at a fork. The override could then be disable when both eyes see white (Logic 1) again. This makes sense to me, but is just speculation at the moment.

The cases where both overrides are active in the truth table can be ignored, I believe, as this scenario should not occur and is redundant (same function as both eyes returning Logic 0).