
PLC 審查問題對應階梯圖

單車道雙向通行控制——15 項問題的修正梯形圖與完整重構程式

對應文件：《PLC 梯形圖程式設計審查報告》(2026-07-05)

內容範圍：問題 1-11 與建議 12-15 之修正階梯圖

附錄：修正後完整程式 Rung 1-23

文件日期：2026 年 7 月 6 日

圖中符號採 IEC 慣用之 NO / NC 接點、線圈與功能方塊表示法；
變數名稱沿用原程式，狀態常數與 I/O 對照見第 1 節。

Contents

1 變數對照與圖例	2
1.1 狀態常數與變數	2
1.2 梯形圖符號圖例	2
1.3 問題與 rung 對照	3
2 嚴重問題修正階梯圖 (問題 1-5)	4
2.1 問題 1：INIT 狀態沒有出口——上電死鎖	4
2.2 問題 2：Rung 24 ——PB_RESET 無條件強制回 IDLE	4
2.3 問題 3：Rung 16 ——REV 的離開條件錯誤	4
2.4 問題 4：計時器 Done 位元未限定狀態	5
2.5 問題 5：邊緣偵測標籤不一致	5
3 重要問題修正階梯圖 (問題 6-11)	7
3.1 問題 6：60 秒逾時應轉入 FAULT，而非回 IDLE	7
3.2 問題 7：REV 狀態不驅動占用指示	7
3.3 問題 8：兩端同時來車 → 雙方永久等待	7
3.4 問題 9：兩支感測器共用一顆卡住偵測計時器	8
3.5 問題 10：故障優先權與復歸順序耦合	9
3.6 問題 11：多車跟進——車輛計數 (選配)	9
4 可維護性建議階梯圖 (建議 12-15)	10
4.1 建議 12：改用內建邊緣接點	10
4.2 建議 13：狀態機集中化	10
4.3 建議 14：計時器與變數命名語義化	10
4.4 建議 15：手動按鈕一律取上升緣	10
5 修正後完整程式 (Rung 1-23)	11
5.1 輸入處理與初始化 (R1-R3)	11
5.2 故障偵測與接管 (R4-R8)	11
5.3 狀態轉移 (R9-R13)	12
5.4 計時與逾時 (R14-R20)	13
5.5 故障復歸與輸出 (R21-R23)	14
5.6 設計要點備忘	14

1 變數對照與圖例

1.1 狀態常數與變數

值	狀態常數	變數	意義
0	INIT	INPUT1 / INPUT2	A 端 / B 端光電感測器
1	IDLE	PB_RESET	故障復歸按鈕
2	FWD	OUTPUT_1 / OUTPUT_2	車道占用燈 / 故障燈
3	HOLD	STATE	狀態暫存器 (INT)
4	REV	M_INPUT1_Rise 等	邊緣旗標 (單一掃描週期)
5	FAULT	FaultLatch / SensorStuck	故障門鎖 / 感測器卡住
6	REV_HOLD (新增)	TON_1/2A/2B/3	10 s 駐留 / 5 min 卡住 / 60 s 逾時

1.2 梯形圖符號圖例

VAR 	NO 常開接點：變數為 1 時導通
VAR 	NC 常閉接點：變數為 0 時導通
VAR 	上升緣接點 (OSR/R_TRIG)：0→1 的那個掃描週期導通一次
VAR 	輸出線圈：rung 導通時 ON，斷開時 OFF
VAR 	設定線圈 (Latch)：導通後保持 ON，需 U 復歸
VAR 	復歸線圈 (Unlatch)：導通時將門鎖變數復歸為 OFF
	比較方塊：條件成立 (相等) 時導通，作為接點使用
	傳送方塊：rung 導通時將來源值寫入目的暫存器
	通電延遲計時器：IN 持續導通達 PT 後 Q = 1；IN 斷開立即復歸

1.3 問題與 rung 對照

問題	摘要	完整程式對應 rung (第 5 節)
問題 1	INIT 無出口 (上電死鎖)	R3
問題 2	Reset 無條件回 IDLE	R21
問題 3	REV 過早結束	R13、R16、R18
問題 4	計時器 Q 未限定狀態	R17、R18
問題 5	邊緣偵測標籤不一致	R1、R2 (改用內建邊緣)
問題 6	逾時應進 FAULT	R14、R15
問題 7	REV 不點占用燈	R22
問題 8	兩端同時來車死鎖	R9-R11
問題 9	卡住偵測共用計時器	R4-R6
問題 10	故障優先權與復歸耦合	R8、R21
問題 11	多車跟進 (車輛計數)	選配 (見 3.6 節)
建議 12-15	內建邊緣、集中化、命名、按鈕取緣	R1、R2、R21 及全篇

2 嚴重問題修正階梯圖（問題 1-5）

2.1 問題 1：INIT 狀態沒有出口——上電死鎖

原程式沒有任何 rung 處理 STATE = INIT（上電預設值 0）。修正：新增一條初始化 rung，等待兩支感測器皆淨空後才進入 IDLE；若上電時感測器被遮斷，系統會停留在 INIT 直到車道淨空，不會誤動作。

修正 新增 rung：INIT 且 A、B 皆淨空 → IDLE



若平台提供首次掃描旗標（如 _SYSVA_FIRST_SCAN），亦可改以首次掃描觸發並在感測器遮斷時直接門鎖故障；上圖的「等待淨空」寫法較簡單且不需額外旗標。

2.2 問題 2：Rung 24 ——PB_RESET 無條件強制回 IDLE

現況 原 Rung 24：按鈕電平直接寫入，任何狀態都會被覆寫



修正 加上三重限定：按鈕取上升緣、僅在 FAULT 狀態、且故障已成功解門



如此車輛通行中誤按復歸不再影響狀態機；故障未排除（FaultLatch 仍為 1）時按住按鈕也不會造成 STATE 每週期震盪。

2.3 問題 3：Rung 16 ——REV 的離開條件錯誤

現況 原 Rung 16：B 下降緣即回 IDLE，此時車輛仍在車道中間



修正：REV 應在車輛抵達 A 感測器時才結束，並比照 FWD 增加 10 秒駐留緩衝狀態 REV_HOLD（新增常數，值 6）。

修正 REV 且 A 上升緣 → REV_HOLD



修正 駐留計時：HOLD 與 REV_HOLD 共用 10 秒計時器 TON_1



修正 REV_HOLD 計時到 → IDLE（狀態限定，見問題 4）



2.4 問題 4：計時器 Done 位元未限定狀態

現況 原 Rung 12 / 14：TON.Q 一成立就無條件寫 IDLE，可能覆寫 FAULT 等狀態



修正 每一條寫入 STATE 的 rung，第一個條件一律是「目前狀態」



原 Rung 12（TON_3.Q 逾時）除了加狀態限定外，目的地也應由 IDLE 改為 FAULT，見問題 6 的階梯圖。

2.5 問題 5：邊緣偵測標籤不一致

現況 原 Rung 1：接點讀 INPUT1_Last，但 Rung 3 線圈寫的是 M_INPUT1_Last ——若為兩個變數，邊緣偵測全失效



修正後三條 rung 全部使用同一個變數 M_INPUT1_Last，且 _Last 的更新 rung 必須排在最後（INPUT2 比照辦理）：

修正 上升緣：本週期為 1、上週期為 0



修正 下降緣：本週期為 0、上週期為 1



修正 上週期值更新——必須放在所有使用 _Last 的 rung 之後



更好的做法是直接改用平台內建的上升緣接點（見建議 12 與第 5 節 R1、R2），六條 rung 縮為兩條，並徹底消除此類命名錯誤。

3 重要問題修正階梯圖（問題 6-11）

3.1 問題 6：60 秒逾時應轉入 FAULT，而非回 IDLE

車輛拋錨卡在車道內超過 60 秒屬於異常；回 IDLE 會放行對向車輛。修正：FWD 與 REV 共用通行逾時計時器，逾時即門鎖故障並轉入 FAULT（占用燈由輸出 rung 在 FAULT 狀態下保持點亮，見問題 7）。

修正 通行逾時計時：FWD 或 REV 期間累計 60 秒



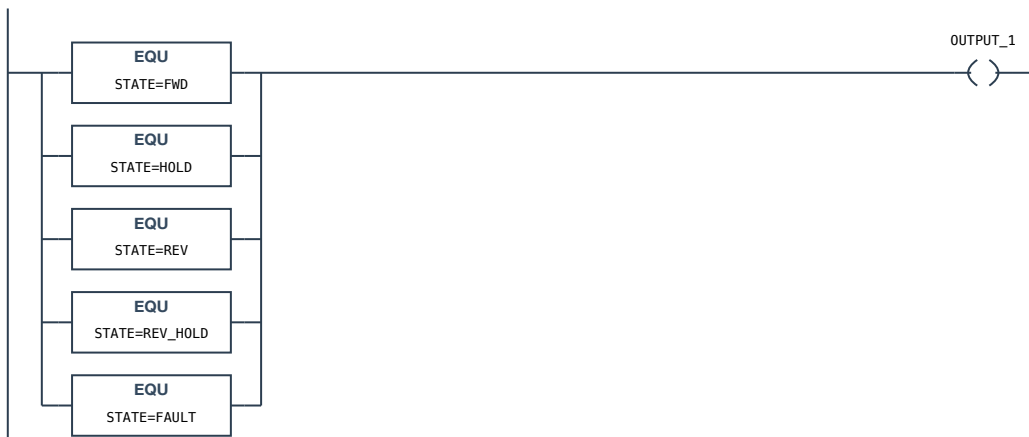
修正 逾時 → 進入 FAULT 並門鎖，需人員確認車道淨空後復歸



3.2 問題 7：REV 狀態不驅動占用指示

原 Rung 17 只有 FWD / HOLD 點亮 OUTPUT_1。修正：REV、REV_HOLD 也並聯進來；FAULT 期間同樣保持點亮（車道狀況未知，必須封鎖）。

修正 占用燈 = 狀態的純函數，五個狀態並聯



3.3 問題 8：兩端同時來車 → 雙方永久等待

原程式以「上升緣 + 對側淨空」觸發，兩端幾乎同時遮斷時兩條 rung 都不成立，且邊緣旗標一個週期後消失，永遠卡死。修正：進入條件改用電平（車輛持續遮斷感測器即成立），並新增「兩側同時佔用」的固定優先權裁決 rung。

修正 IDLE → FWD：A 遮斷（電平）且 B 淨空



修正 IDLE → REV：B 遮斷（電平）且 A 淨空



修正 同時佔用的仲裁：固定 A 側優先（依現場規則亦可改為進入 FAULT 或輪流優先）



注意 rung 排列順序：兩條單側 rung 在前、仲裁 rung 在後；三條的條件互斥，同一週期至多一條成立。

3.4 問題 9：兩支感測器共用一顆卡住偵測計時器

現況 原 Rung 19：INPUT1 OR INPUT2 餵同一顆 TON，計時跨感測器連續累積而誤報



修正：每支感測器各用一顆計時器，且僅在與狀態矛盾時計時（IDLE 卻長時間遮斷才可疑；通行中的合法遮斷不計）。

修正 A 端獨立卡住偵測



修正 B 端獨立卡住偵測



修正 任一計時到即判定感測器卡住



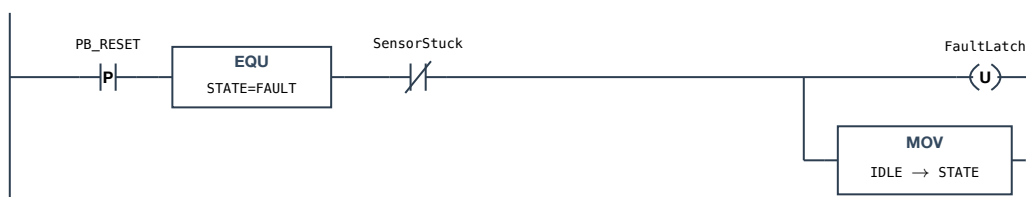
3.5 問題 10：故障優先權與復歸順序耦合

修正有兩點：(1) 「FaultLatch → FAULT」 rung 移到狀態機之前，故障成立的同一週期即接管狀態；(2) 解門與回 IDLE 合併為單一 rung，不再依賴三條 rung 的掃描順序。

修正 故障接管——置於所有狀態轉移 rung 之前



修正 復歸整合：解門與回 IDLE 同一 rung 完成



3.6 問題 11：多車跟進——車輛計數（選配）

若場域允許多車同時在車道內，以「車道內車輛數」取代純計時判斷，以下以 FWD 方向為例（REV 鏡像對稱）：

修正 A 端進入一台，計數 +1（邊緣觸發，避免每週期累加）



修正 B 端離開一台，計數 -1



修正 回 IDLE 的條件加上「車道內車輛數 = 0」



車輛計數屬功能擴充，第 5 節的完整程式未納入；導入時另需處理 VehCnt 的下限保護與 FWD 期間後車跟進重啟逾時計時等細節。

4 可維護性建議階梯圖（建議 12–15）

4.1 建議 12：改用內建邊緣接點

手寫三段式邊緣偵測（6 條 rung、2 個中間變數）以內建上升緣接點（OSR / R_TRIG）取代，每支感測器一條 rung 即可，並徹底避免問題 5 的命名斷裂：

修正 一條 rung 取代原 Rung 1–3



4.2 建議 13：狀態機集中化

梯形圖版本的落地原則（完整落實見第 5 節）：

- 每個 MOV → STATE 一律先串 EQU(STATE, 現態)；
- 相同現態的多個轉移條件彼此互斥；
- 故障接管 rung 放在所有轉移之前、輸出 rung 放在最後。

若平台支援，狀態轉移改用 ST 的 CASE 集中管理更佳（見審查報告第 3.2 節）。

4.3 建議 14：計時器與變數命名語義化

原名	建議名	原名	建議名
TON_1	tHoldDwell (10 s)	OUTPUT_1	LaneOccupiedLamp
TON_2A/2B	tStuckA / tStuckB (5 min)	OUTPUT_2	FaultLamp
TON_3	tTransitTimeout (60 s)	INPUT1/2	SensorA / SensorB

修正 命名語義化後的同一條 rung，用途一目瞭然



4.4 建議 15：手動按鈕一律取上升緣

已落實於問題 2 / 問題 10 的復歸 rung（PB_RESET 使用 P 接點）：按住不放只觸發一次，不會每個掃描週期重複執行 MOV。

5 修正後完整程式 (Rung 1-23)

以下為整合所有修正後的完整梯形圖程式。結構依「輸入處理 → 故障偵測 → 狀態轉移 → 計時 → 復歸 → 輸出」排列，STATE 的每個寫入點都以現態為第一條件，輸出為狀態的純函數。

5.1 輸入處理與初始化 (R1-R3)

R1 INPUT1 上升緣 (內建邊緣接點，取代原 Rung 1-3)



R2 INPUT2 上升緣 (取代原 Rung 4-6)



R3 初始化：INIT 且車道淨空 → IDLE (修正問題 1)



5.2 故障偵測與接管 (R4-R8)

R4 A 端卡住偵測：IDLE 卻持續遮斷 5 分鐘 (修正問題 9)



R5 B 端卡住偵測



R6 任一計時到 → SensorStuck



R7 卡住即門鎖故障**R8** 故障接管——置於狀態機之前（修正問題 10）**5.3 狀態轉移（R9–R13）****R9** IDLE → FWD : A 遮斷（電平）且 B 淨空（修正問題 8）**R10** IDLE → REV : B 遮斷（電平）且 A 淨空**R11** 同時佔用仲裁：A 側優先（修正問題 8）**R12** FWD → HOLD : 車輛通過 B**R13** REV → REV_HOLD : 車輛抵達 A（修正問題 3）

5.4 計時與逾時 (R14-R20)

R14 通行逾時計時：FWD 或 REV 共用 60 秒（修正問題 6、7 之計時部分）



R15 逾時 → FAULT 並門鎖（修正問題 4、6）



R16 駐留計時：HOLD 或 REV_HOLD 共用 10 秒



R17 HOLD 駐留期滿 → IDLE（狀態限定，修正問題 4）



R18 REV_HOLD 駐留期滿 → IDLE



R19 HOLD 期間後車自 A 跟進 → 回 FWD

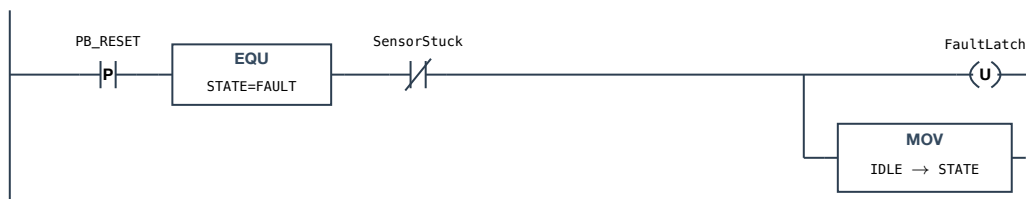


R20 REV_HOLD 期間後車自 B 跟進 → 回 REV

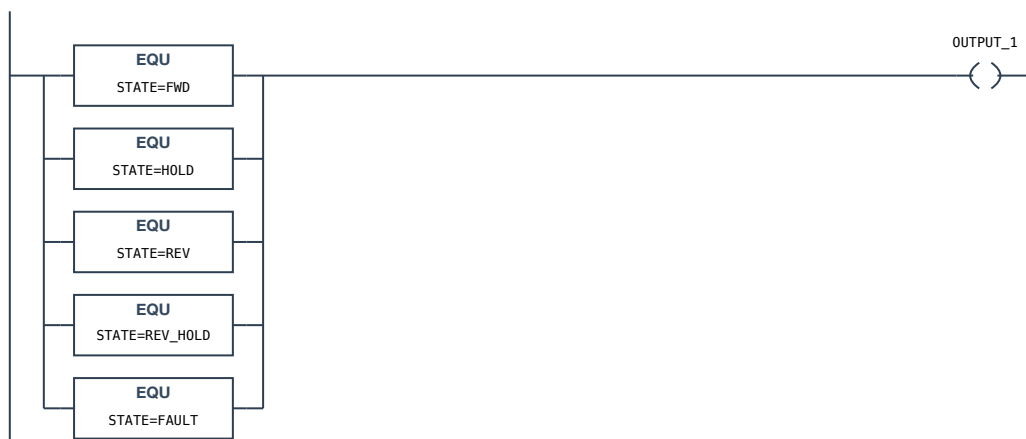


5.5 故障復歸與輸出 (R21-R23)

R21 復歸：按鈕上升緣、限 FAULT 狀態、故障已排除；解門與回 IDLE 一次完成（修正問題 2、10、建議 15）



R22 占用燈：通行、駐留與故障期間皆點亮（修正問題 7）



R23 故障燈



5.6 設計要點備忘

- **寫入點限定：**STATE 的寫入 rung (R3、R8-R13、R15、R17-R21) 除故障接管 (R8) 與逾時 (R15，由狀態驅動的計時器保證) 外，全部以 EQU(STATE, 現態) 為第一條件，同一現態的轉移條件互斥。
- **掃描順序：**邊緣旗標 (R1-R2) 最先產生；故障接管 (R8) 在所有轉移之前；輸出 (R22-R23) 在最後，為狀態的純函數。
- **計時器：**全部使用 TON（非保持型），IN 由狀態驅動，離開狀態即自動復歸；R15、R17、R18 使用 TON.Q 時皆有狀態限定，即使日後搬動 rung 也不會誤寫。
- **逾時即故障：**60 秒逾時不再放行對向車輛，一律門鎖 FAULT，由人員確認車道淨空後以 R21 復歸。
- **選配：**多車場域請加入第 3.6 節的車輛計數，以「車數 = 0」取代純計時作為回 IDLE 的條件。