設計書

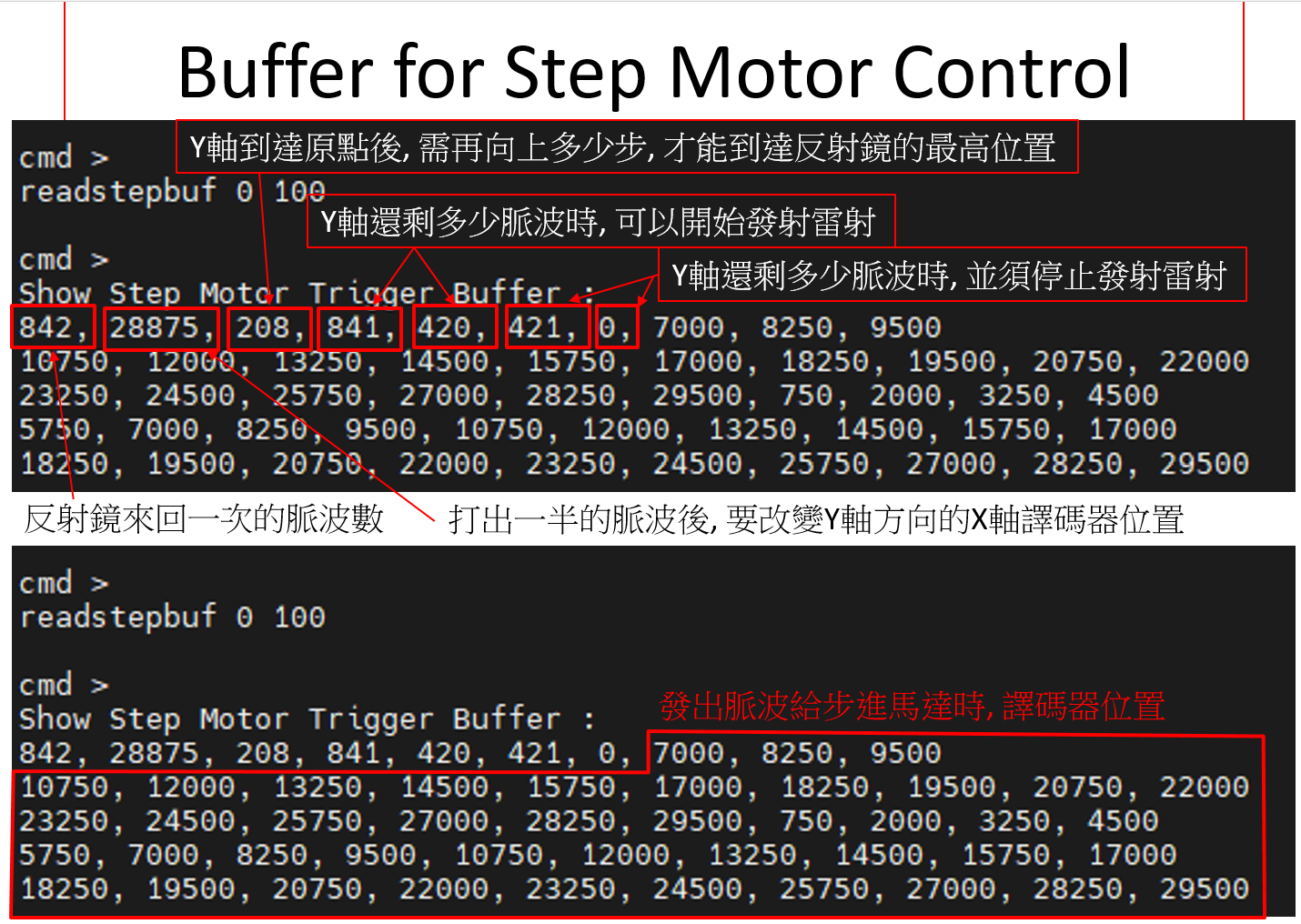
修改

規格 :

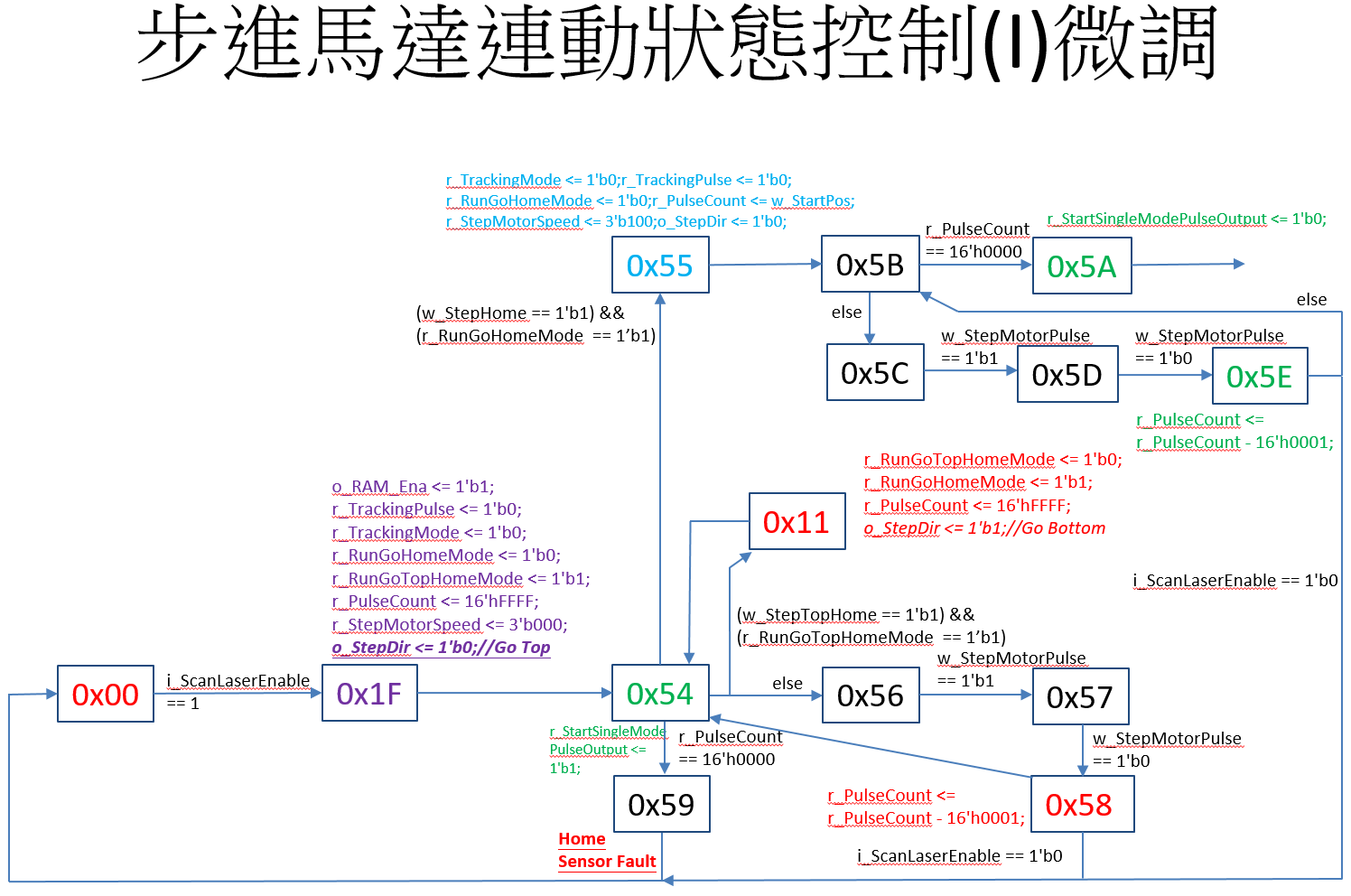
1. 水平 600 點, 角解析度 0.1度
2. 垂直 300條線, 角解析度 0.1度
3. 雷射重複率 : 大於 100KHz
4. 距離解析度 : 1 cm
5. 測距最大距離 : > 500M
6. 結束

設計說明 :

1. 水平角解析度為 0.1度, 雷射重複率為100KHz, 所以BLDC 每10usec轉動0.05度, 因此BLDC轉動一圈的時間為 (360/0.05)\*10usec = 72000usec = 72msec, 得到BLDC每秒轉動13.89 rps = 833 rpm
2. 垂直300條線, 每四條線需要72msec(每條線18msec), 所以掃描300條線至少須要5.4sec, 因此 Frame Rate = 0.185 fps
3. 垂直角解析度為0.1度, 微步進驅動器的角解析度為 1.8/256 = 0.007031度(microstepping indexer = 256), 所以BLDC轉動90度, 步進馬達需要走14.2222步, 最大控制角度誤差為 0.003516度(實際角度誤差可能更大)
4. 步進馬達控制範例 :



1. 步進馬達運作說明 :
   1. 步進馬達先讓反射鏡向上壓到上方極限, 再讓反射鏡向下壓到下方極限
   2. 步進馬達讓反射鏡向上達到最高點, 然後再開始讓反射鏡來回掃描
2. 反射鏡控制設計(Buffer for Step Motor Control) :
   1. 反射鏡來回一次的脈波數 : 8534
   2. 打出一半的脈波後(反射鏡掃到下方), 需要改變Y軸方向的X軸Encoder位置
   3. Y軸到達原點後,步進馬達需要再向上走多少步,才能到達反射鏡正常工作的最高位置
   4. Y軸步進馬達還剩下多少步(脈波)時,可以開始發射雷射(反射鏡掃到上方)
   5. Y軸步進馬達還剩下多少步(脈波)時,可以開始發射雷射(反射鏡掃到下方)
   6. Y軸步進馬達還剩下多少步(脈波)時,必須停止發射雷射(反射鏡掃到上方)
   7. Y軸步進馬達還剩下多少步(脈波)時,必須停止發射雷射(反射鏡掃到下方)
3. 步進馬達控制原理說明 : 步進馬達啟動掃描時, 先讓轉鏡向上翻轉至上極限, 然後再讓轉鏡向下翻轉至下極限, 最後再將轉鏡向上翻轉至掃描的最高點後, 開始進行掃描



設定轉鏡翻轉至上極限

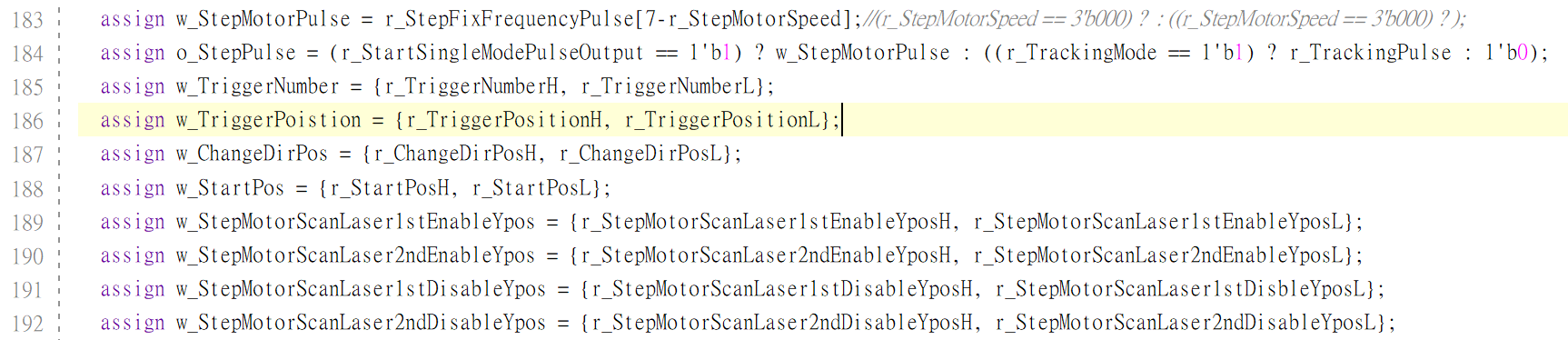
設定轉鏡翻轉至下極限

轉鏡翻轉壓到上極限

轉鏡翻轉壓到下極限

啟動步進馬達追蹤程序前, 必須從 Block Memory中讀取追蹤控制的相關參數

由狀態 0x54 到 0x5A的過程中有做微幅修改

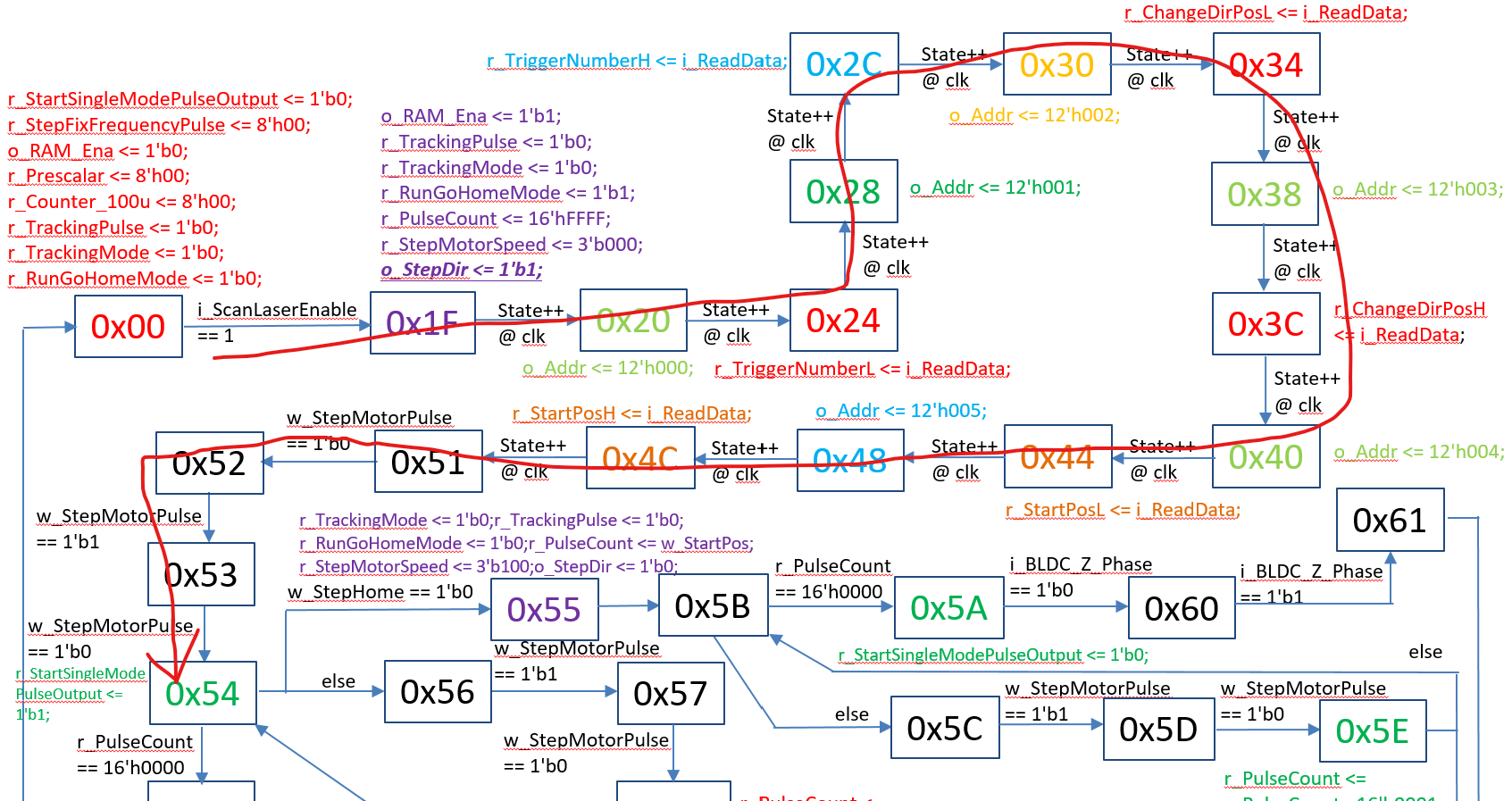


r\_StepFixFrequencyPulse : 是一個 8 bit counter, clock source 的頻率為10KHz, 因此這個 Counter 的每個bit可以產生5KHz, 2.5KHz, 1.25KHz, 625Hz, 312.5Hz, 156.25Hz, 78.125Hz, 39.063Hz

r\_StepMotorSpeed : 可設定0-7, 選擇上面八個頻率的一個由 w\_StepMotorPulse 輸出

實際輸出到步進馬達的Pulse 是由 o\_StepPulse送出, 當在單動模式下(r\_StartSingleModePulseOutput == 1), o\_StepPulse接至w\_StepMotorPulse, 當在追蹤模式下(r\_StartSingleModePulseOutput == 0 且 r\_TrackingMode == 1), o\_StepPulse接至r\_TrackingPulse

r\_TrackingPulse : 根據Lookup Table記錄的Encoder數值,產生追蹤Encoder位置的脈波



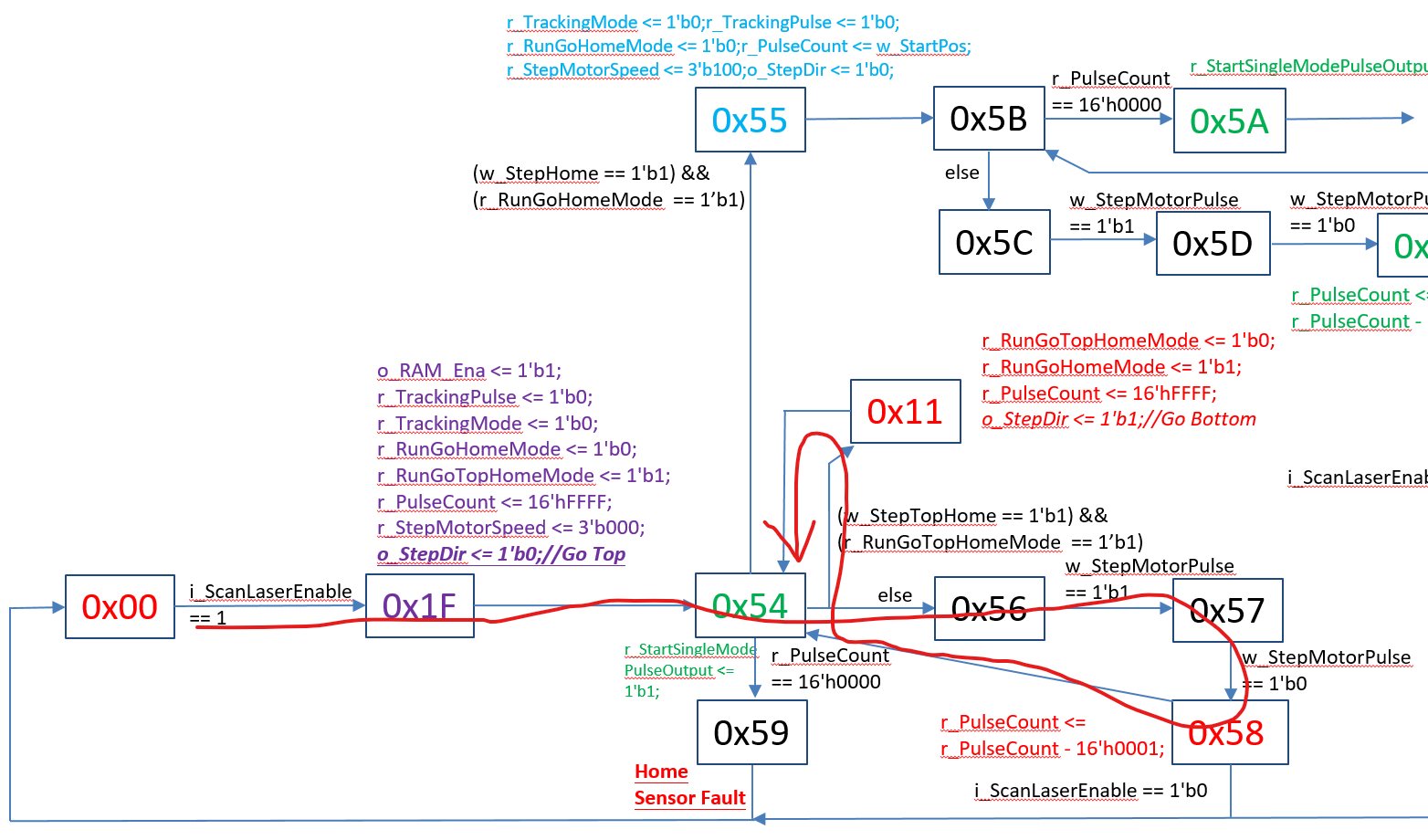
狀態 0x20到0x4C 會執行讀取以下參數的程序 :

讀取 w\_TriggerNumber = {r\_ TriggerNumberH, r\_ TriggerNumberL}

讀取 w\_ChangeDirPos = {r\_ ChangeDirPosH, r\_ ChangeDirPosL}

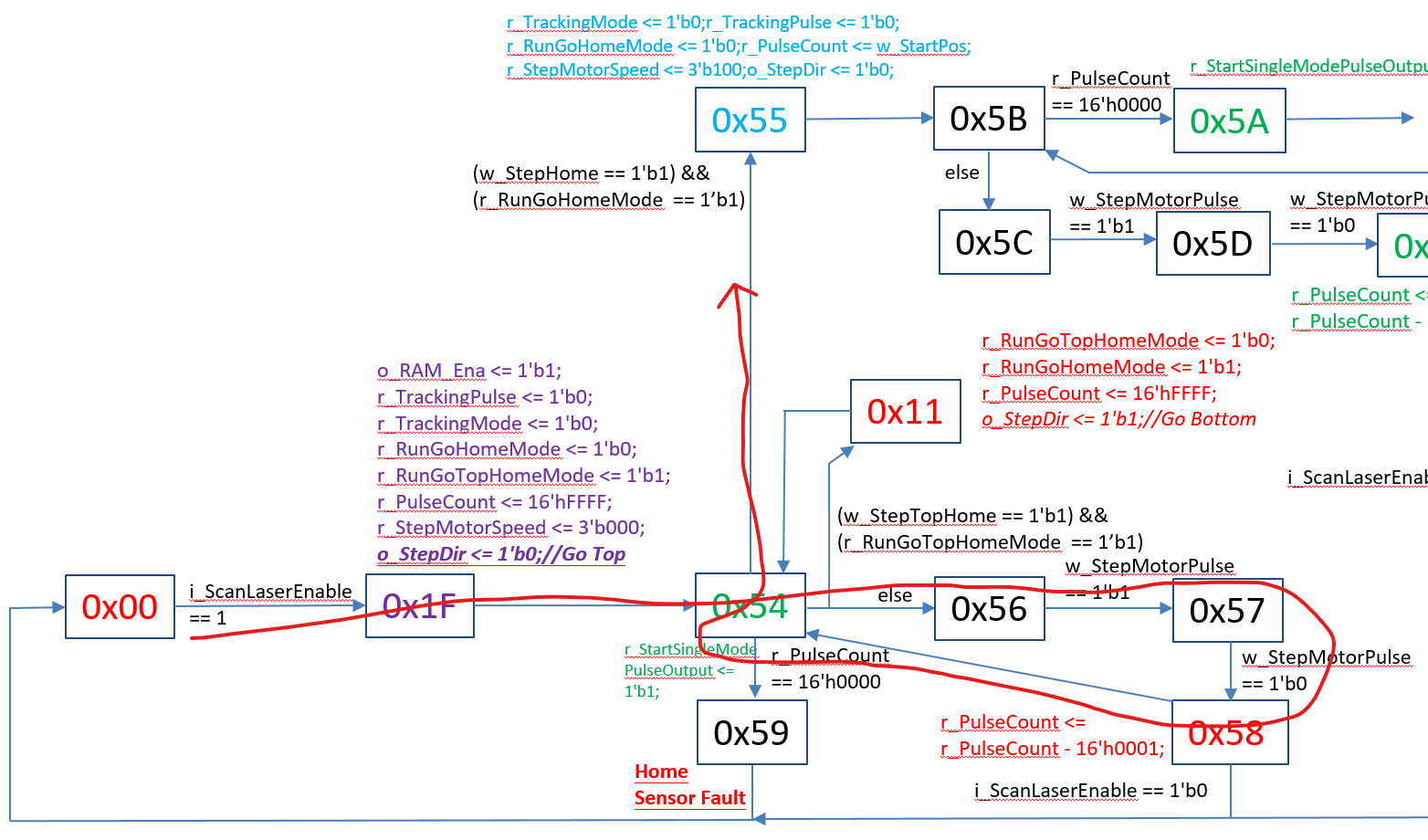
讀取 w\_StartPos = {r\_ StartPosH, r\_ StartPosL}

r\_TriggerNumberL : 存放在 address = 0

r\_StartPosH : 存放在 address = 5 

Go Top Home 的流程

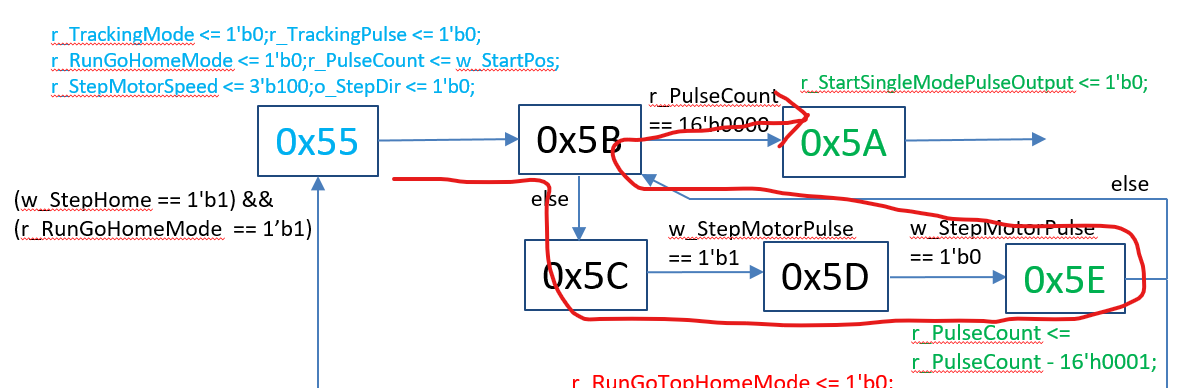
當 w\_StepTopHome == 1 且在 r\_RunGoTopHomeMode == 1 時, 會進入 0x11 的狀態, 將 r\_RunGoTopHomeMode 設為 0 並設定r\_RunGoHomeMode = 1, 停止 Go Top Home Mode 並啟動Go Home Mode



Go Home 的流程

當 w\_StepHome == 1 且在 r\_RunGoHomeMode == 1 時, 會進入 0x55 的狀態, 將 r\_RunGoHomeMode 設為 0, 停止 Go Home Mode, 同時並設定前往起點需要的脈波數(r\_PulseCount <= w\_StartPos), 然後進入 0x5B狀態, 開始進行前往掃描起點的程序

步進馬達將轉鏡轉至由上往下掃描起點的流程 : 當轉鏡轉至由上往下掃描起點後, 就會開始進行垂直方向的掃描(步進馬達讓轉鏡上下掃描), 到達起點後, 要結束固定頻率的脈波輸出, 因此將 r\_StartSingleModePulseOutput 設為 0, 所以, 將r\_StartSingleModePulseOutput 設為 1'b0, 此後, 步進馬達要根據無刷馬達的位置, 決定轉鏡是否需要轉動(轉鏡需要轉動)



狀態 0xC0到0xFF 會執行讀取以下參數的程序 :

讀取 w\_StepMotorScanLaser1stEnableYpos =

{r\_ StepMotorScanLaser1stEnableYposH, r\_ StepMotorScanLaser1stEnableYposL}

讀取 w\_StepMotorScanLaser2ndEnableYpos =

{r\_StepMotorScanLaser2ndEnableYposH, r\_StepMotorScanLaser2ndEnableYposL}

讀取 w\_StepMotorScanLaser1stDisableYpos =

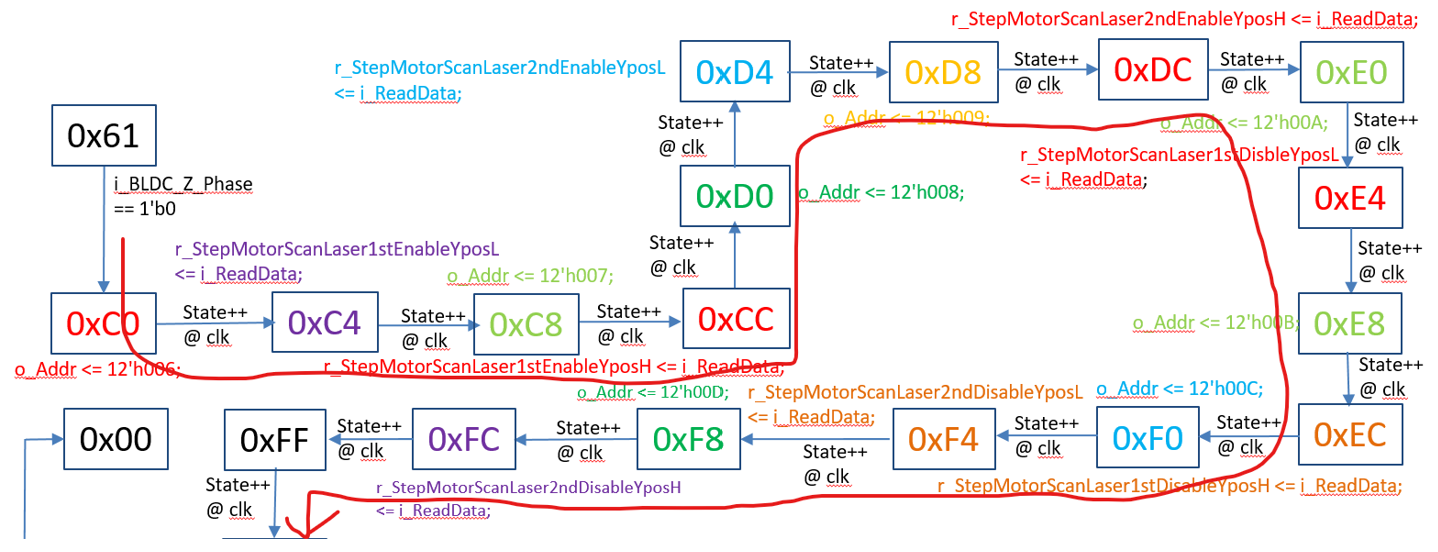
{r\_StepMotorScanLaser1stDisableYposH, r\_StepMotorScanLaser1stDisbleYposL};

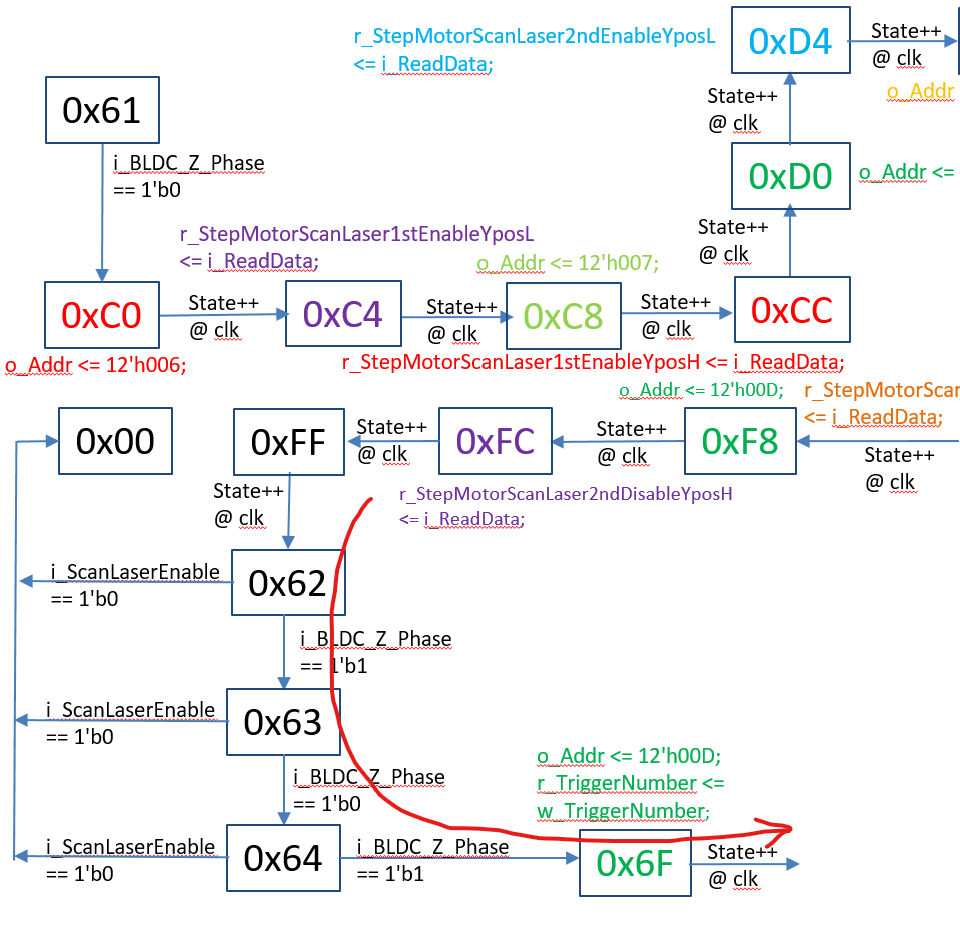
讀取w\_StepMotorScanLaser2ndDisableYpos =

{r\_StepMotorScanLaser2ndDisableYposH, r\_StepMotorScanLaser2ndDisableYposL};

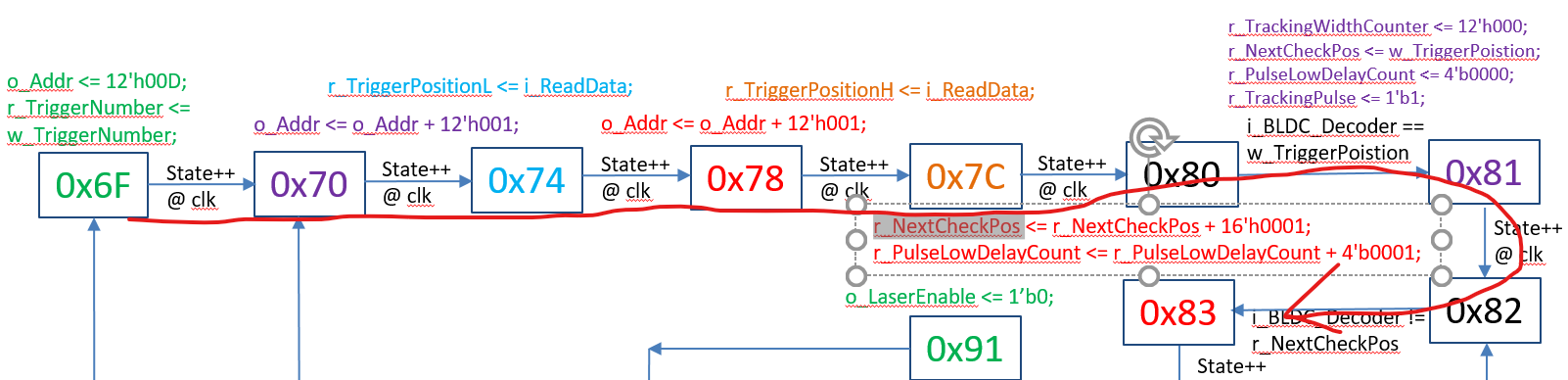
r\_ StepMotorScanLaser1stEnableYposL : 存放在 Block Memory 的 address = 6

r\_StepMotorScanLaser2ndDisableYposH : address = 存放在 Block Memory 的 13(0x0D)



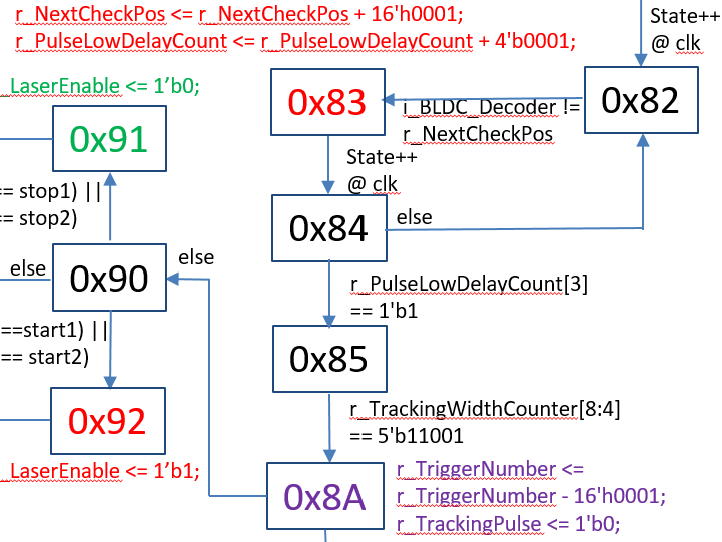


所有目前需要的參數都載入後, 會等候兩次 Z-phase 後(Z-phase = 1 🡺 0 🡺 1), 進入追蹤掃描程序(由狀態 0x6F進入)

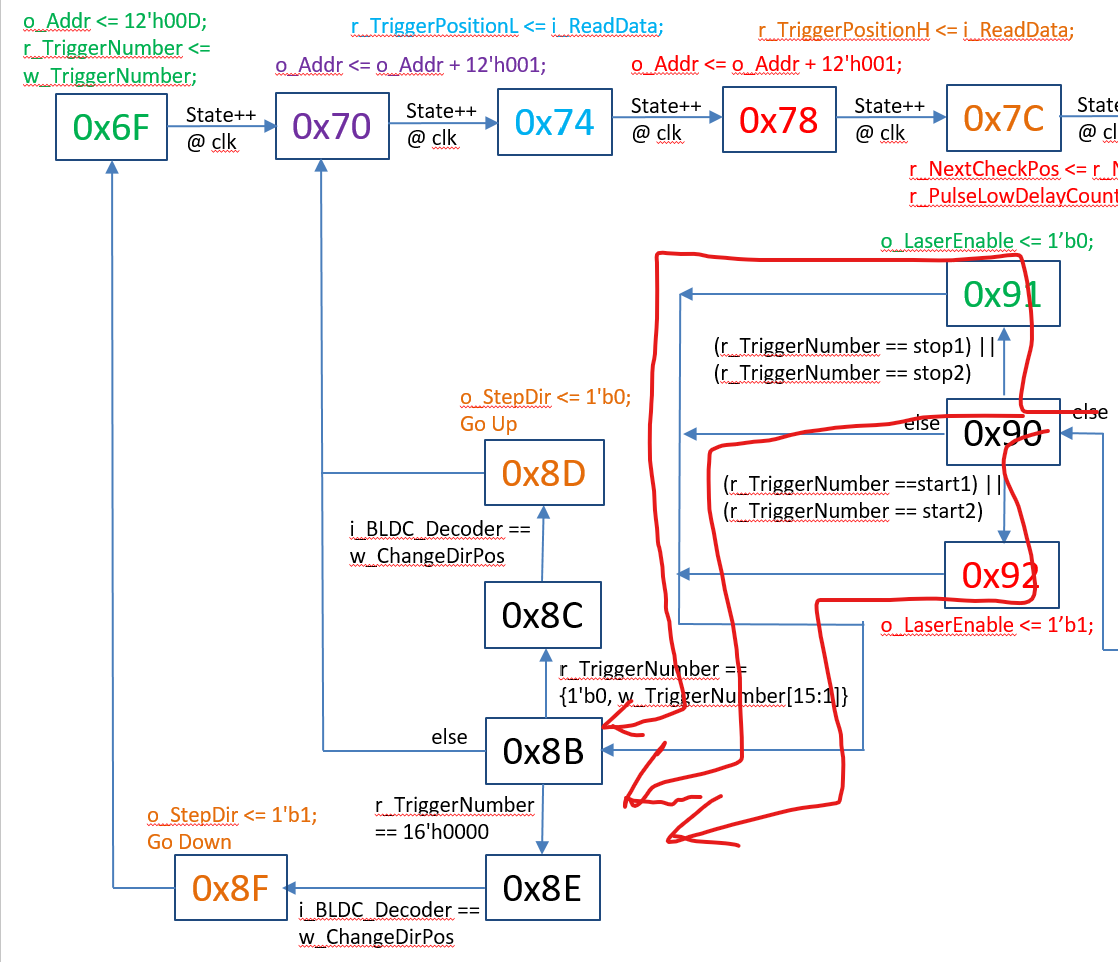


狀態 0x74時會載入r\_TriggerPositionL, 壯代0x7C時會載入 r\_ TriggerPositionH (w\_TriggerPosition = {r\_ TriggerPositionH , r\_ TriggerPositionL})

當無刷馬達的Encoder到達w\_ TriggerPosition位置時, r\_TrackingPulse設Hi, 同時將 r\_NextCheckPos 設為 r\_TriggerPosition位置



並等候無刷馬達的Encoder離開r\_ NextCheckPos 後, 才往下一個狀態(0x83), 每個Encoder數值變化時, r\_PulseLowDelayCount都會加1, 直到r\_PulseLowDelayCount >= 8後, 再去確認 r\_TrackingWidthCounter >= 400(0x19x)是否成立, 條件確立後, 進入狀態0x8A, 將 r\_TriggerNumber減1及r\_TrackingPulse設為0, 結束一個步進馬達的脈波控制

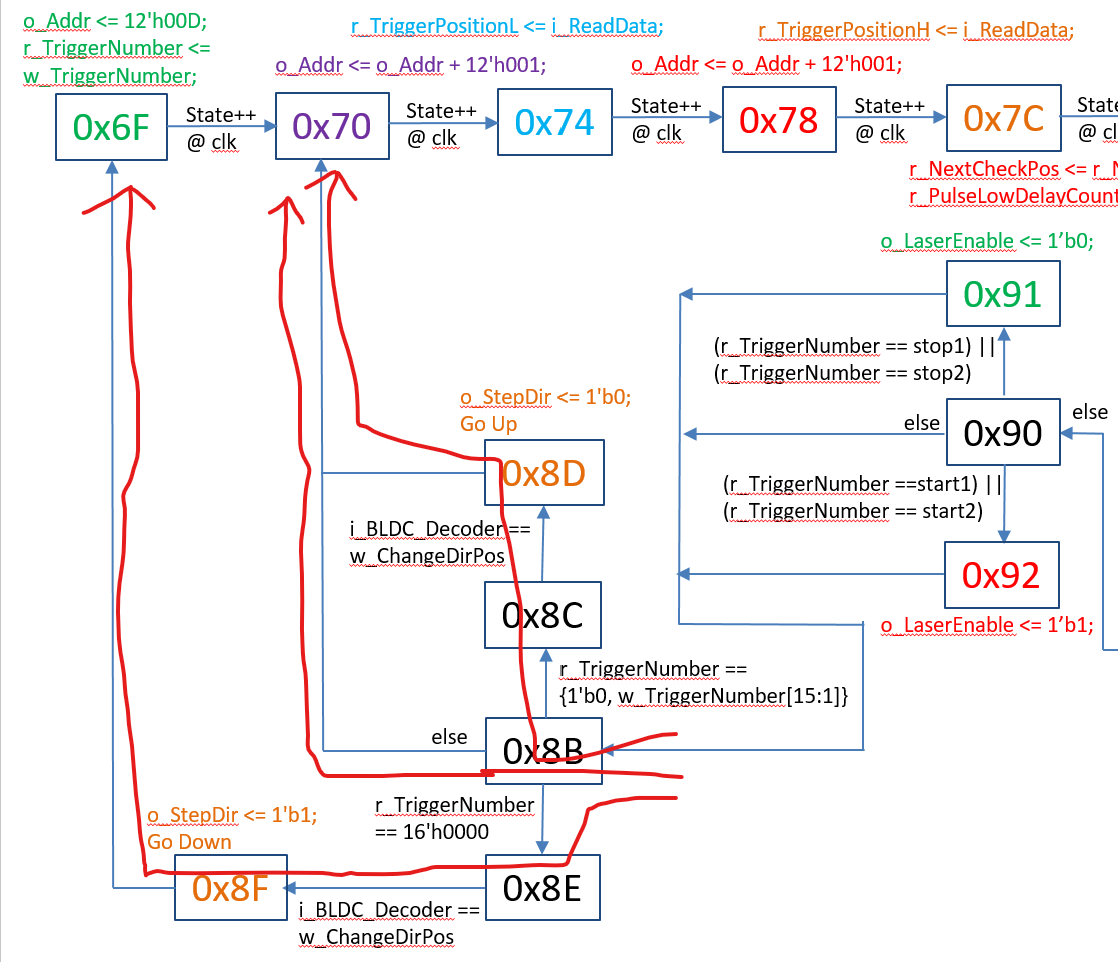


每完成一個步進馬達的脈波, r\_TriggerNumber 會減一, 若r\_TriggerNumber == stop1 或 stop2, 則 o\_LaserEnable <= 0, 若r\_TriggerNumber == start1 或 start2, 則 o\_LaserEnable <= 1, 否則 o\_LaserEnable狀態不變

當無刷馬達轉到發射角度時,只有o\_LaserEnable == 1時, Laser 才能發射

start1 和 start2 分別為 : w\_StepMotorScanLaser1stEnableYpos和 w\_StepMotorScanLaser2ndEnableYpos

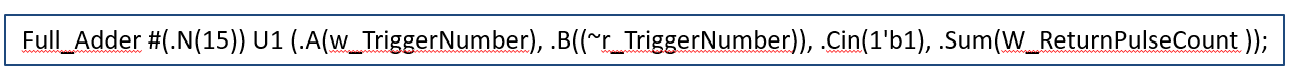
stop1 和 stop2 分別為 : w\_StepMotorScanLaser1stDisableYpos 和 w\_StepMotorScanLaser2ndDisableYpos

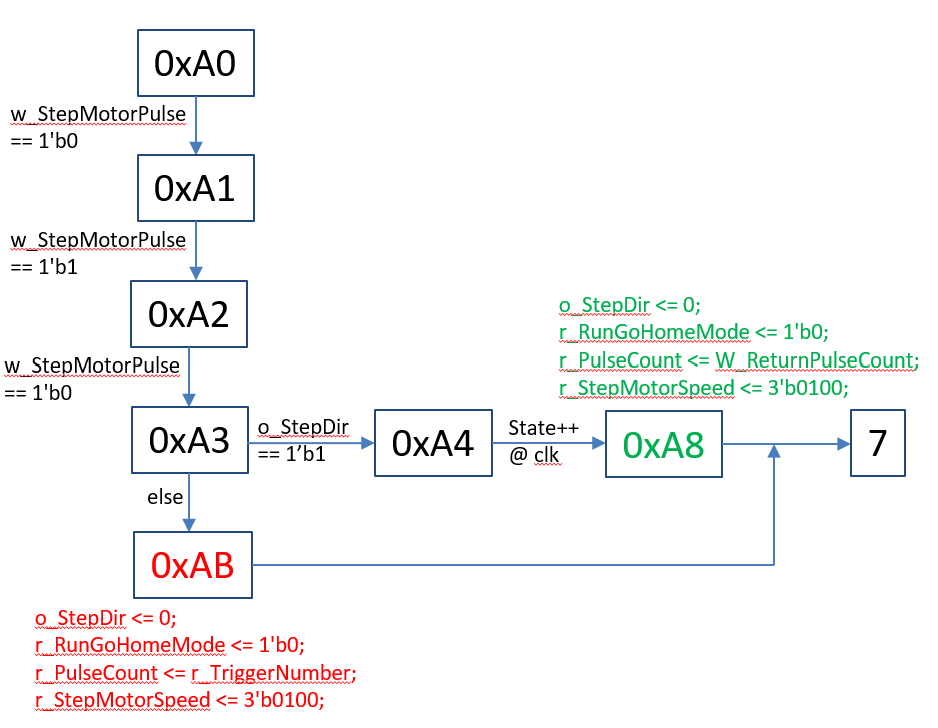


當r\_TriggerNumber == w\_TriggerNumber(存放在 Block Memory的 0x00和0x01的位置) 的一半時, 或 r\_TriggerNumber == 0 時, 在無刷馬達Encoder為w\_ChangeDirPos(存放在 Block Memory的 0x02和0x03的位置)時, o\_StepDir的狀態才能改變,

當r\_TriggerNumber == w\_TriggerNumber時, o\_StepDir <= 0, 步進馬達讓轉鏡向上, 步進馬達控制狀態 0x70, 繼續讀取下一個 r\_TriggerPosition

當r\_TriggerNumber == 0時, o\_StepDir <= 1, 步進馬達讓轉鏡向下, 步進馬達控制狀態 0x6F, 將r\_TriggerNumber設為w\_TriggerNumber, o\_Addr = 0x0D, 從頭開始讀取 r\_TriggerPosition(第一個r\_TriggerPosition)





步進馬達控制狀態0xA0這部分的流程(如上圖), 目前沒有設計進 Verilog 中

這部分的控制是, 等w\_StepMotorPulse == 0 時, 無論步進馬達是正在讓轉鏡向下轉至低點或向上轉至高點, 都是設為向上轉至高點(o\_StepDir均設為0)

1. 測距最大距離為 500M = 50000cm, 所以可使用2 Bytes來表示距離, 則水平600點須使用1200 Bytes的空間來存放距離資訊, 此長度低於 1458 Bytes(一個UDP封包的最大資料容量), 可使用一個封包傳送
2. 若最大距離超過655.35M = 65535cm, 則需使用4 Bytes來表示距離, 因此水平600點需要使用2400 Bytes的空間來存放距離資訊, 此長度高於 1458 Bytes(一個UDP封包的最大資料容量), 須拆成兩個封包傳送
3. 封包格式 :
   1. Header(2 Bytes) : 0xAA55
   2. Echo 訊息(2 Bytes) :
      1. 掃描線號碼(bit0 – bit8): 0-299
      2. Echo ID(bit9 - bit12) : 強度, 掃描線的前半部, 掃描線的後半部, 整條掃描線
      3. Echo 位置(bit13 – bit15) : 0 – 4 表示 1st Echo – 5th Echo
   3. 額外訊息(2 Bytes) : Frame ID
4. 結束