

威綸科技股份有限公司

烤箱溫度控制

FB PID 應用

工程檔案範例

目錄

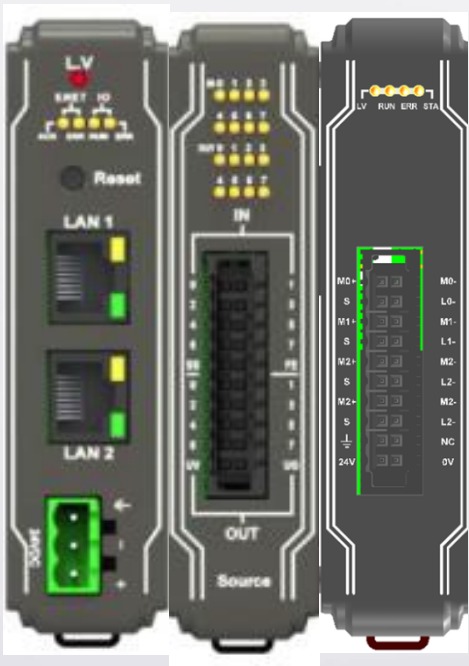
| | |
|--------------------------------|---|
| 1. 簡介與系統運作..... | 1 |
| 2. Weintek Library 功能塊介紹 | 3 |
| 3. 設定程序..... | 5 |
| 4. 控制曲線圖..... | 7 |

1. 簡介與系統運作

簡介

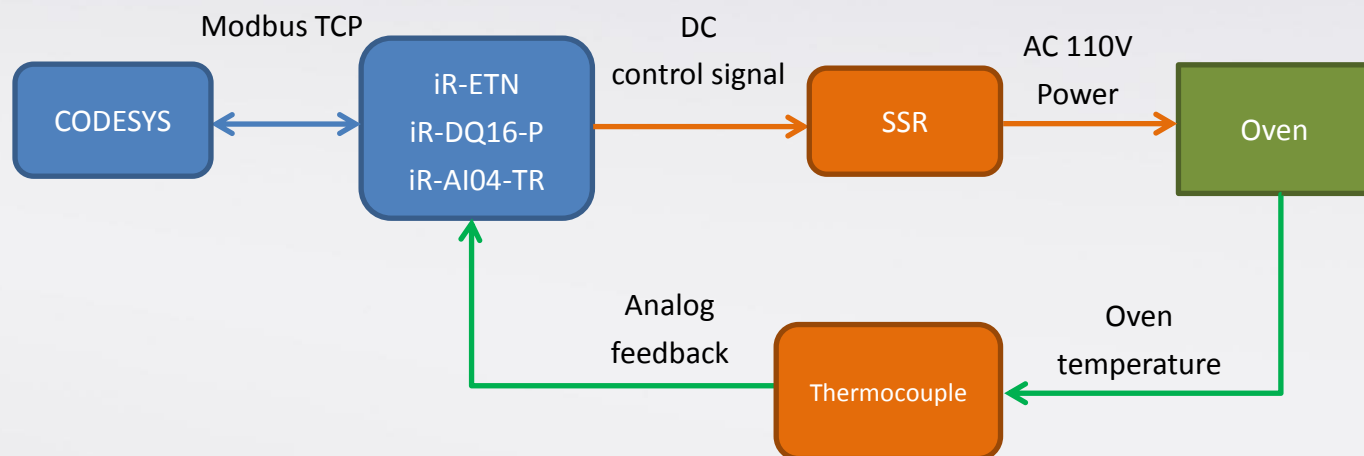
以下範例介紹如何使用 SSR 固態繼電器控制 AC 電源達到 PWM 信號作溫度控制，並利用威綸自行開發的 PID 功能塊進行自動校正。

系統運作



| | | | |
|----|--------|-----------|------------|
| 順序 | 0 | 1 | 2 |
| 模組 | iR-ETN | iR-DM16-P | iR-AI04-TR |
| 名稱 | | | |

烤箱信號流程圖：



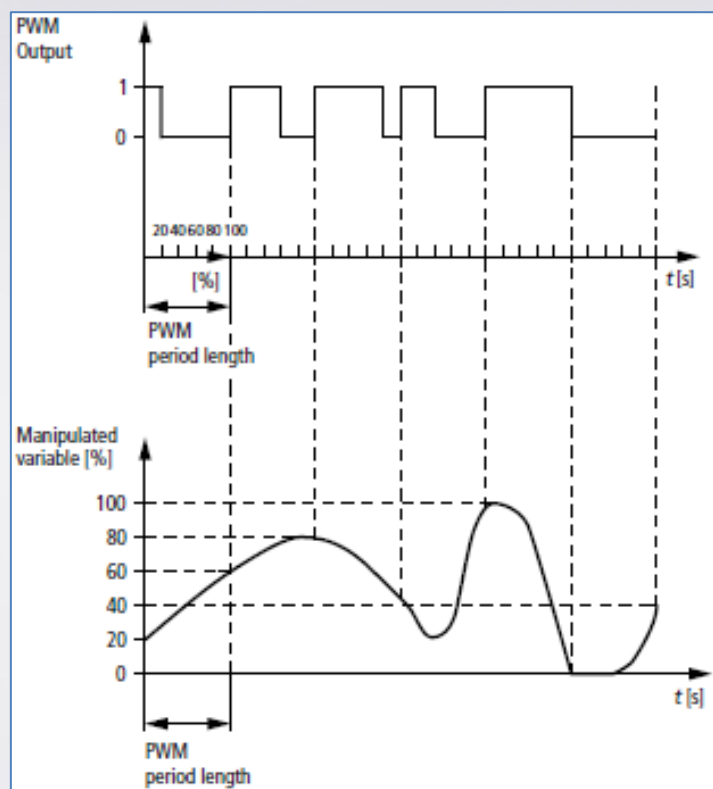
CODESYS 通過乙太網路連接 iR-ETN 控制 iR-DQ16-P 的數位輸出信號。

iR-DQ16-P 數位輸出控制交流電源 AC110V 的導通時間。

藉由熱電偶(Thermocouple)回授溫度數值到 iR-AI04-TR 由 PID 判斷下次導通週期。

※PWM (Pulse Width Modulation：脈波寬度調變)：

是將 PID 計算後的輸出依脈波輸出，在此應用的 PID 計算結果為一個脈波周期的 Duty cycle。



2. Weintek Library 功能塊介紹

如何加入 Weintek Library 請參考文件《威綸函式庫說明》的第二章：

UM018017T_CODESYS_Weintek_Library_UserManual_20190305_cht

PID

威綸自行開發的 PID 控制器功能塊，搭配類比模組進行 PID 控制。此 PID 功能塊的相關公式可參閱前述《威綸函式庫說明》手冊，而本工程範例文件主要講述如何使用。

PID 的輸出方式分為手動/自動兩種，在手動模式下($Manual=TRUE$)，輸出(MV)等於手動輸出值(Mout)，其他參數不對輸出數值造成影響。

自動模式下($Manual=FALSE$)，則依照 PID 控制方程式的參數自動輸出。影響自動輸出的參數有 K_p 、 K_i 、 K_d 、 T_f 、BIAS、Time_Base、Error_Deadband 參數。

若不清楚 PID 參數該如何輸入，我們也提供 PID 自動調試(Autotune)的功能，在

Run & Autotune = TRUE 的情況下開啟。開啟後出全力(MV=MV_Max)直到 PV 到達 SV，接著釋放(MV=MV_Min)等待 PV 小於 SV 後再出全力當 PV 第二次等於 SV 時，Autotune 完成(Autotune=FALSE)即 Kp、Ki、Kd、Tf 調適完成，PV 曲線可參考《4. 溫控曲線》的 Autotune 區段。

PWM

使用 PWM(Pulse Width Modulation)信號的功能塊。輸入 PWM 的週期(Period)與責任比例(Duty)，輸出(Q)從 TRUE 開始輸出。

舉例若要輸出一個周期為 10 秒，其中 Q=TRUE 的時間為 2.5 秒的 PWM 信號，Period = T#10S ； Duty = 25

3. 設定程序

宣告

```

1  PROGRAM PLC_PRG
2  VAR
3      PID : weintek.PID;
4      xRun, xDir, xQE, xDE, xAutoTune : BOOL ;
5      rSV, rPV, rAuto_Deadband, rBIAS, rMV, rI_MV : REAL ;
6      rTime_Base : REAL := 0.1 ;
7      rError_Deadband : REAL := 0.0 ;
8      rMV_Max : REAL := 100.0 ;
9      rMV_Min : REAL := 0.0 ;
10     PWM : weintek.PWM ;
11     rTemp : REAL ;

```

宣告程式中必要的變數並給定初值。

ST 程式

```

1  rTemp2 :=INT_TO_REAL(TR0) ;
2  rPV := rTemp2 / 10 ;
3  PID(
4      Manual:= xManual,
5      Run:= xRun,
6      SV:= rSV,
7      PV:= rPV,
8      Dir:= xDir,
9      MV_Manual:= rMV_Manual,
10     MV_Max:= rMV_Max,
11     MV_Min:= rMV_Min,
12     Auto_Deadband:= rAuto_Deadband,
13     Bias:= rBIAS,
14     Time_Base:= rTime_Base,
15     Error_Deadband:= rError_Deadband,
16     MV=> rMV,
17     I_MV=> rI_MV,
18     Kp:= PVar.rKp,
19     Ki:= PVar.rKi,
20     Kd:= PVar.rKd,
21     Tf:=PVar.rTf,
22     AutoTune:= xAutoTune);
23
24     PWM(Enable:=xRun,Period:=timPeriod,Duty:=rMV,Q=> xOut);
25     DO0 := xOut ;
26

```

1~2：溫度輸入為 INT，轉換單位=> REAL

3~22：PID 功能塊帶入變數

24~25：將 PID.MV 轉化為 PWM 輸出時間由 DO0 輸出

Mapping

DO0 => %QX0.0（控制 SSR 的輸出）

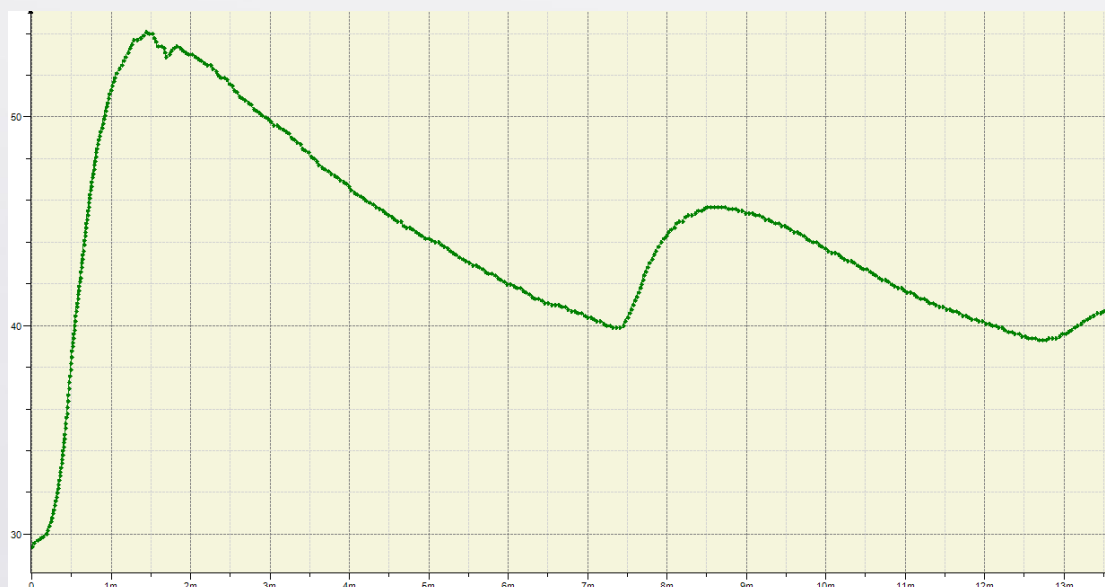
TR0 => %IW0（溫度輸入）

4. 控制曲線圖

範例實際操作的溫控曲線如下，綠線為溫度，起始溫度為常溫約 28°C ，目標溫度 SV 等於 40°C 。

Auto Tuning 溫度曲線

一開始先作 Auto Tune 抓出 K_p 、 K_i 、 K_d 、 T_f 的參考數值，下圖為 Auto Tune 期間溫度曲線。



Auto Tune 流程：

Full power 直到溫度達到 SV

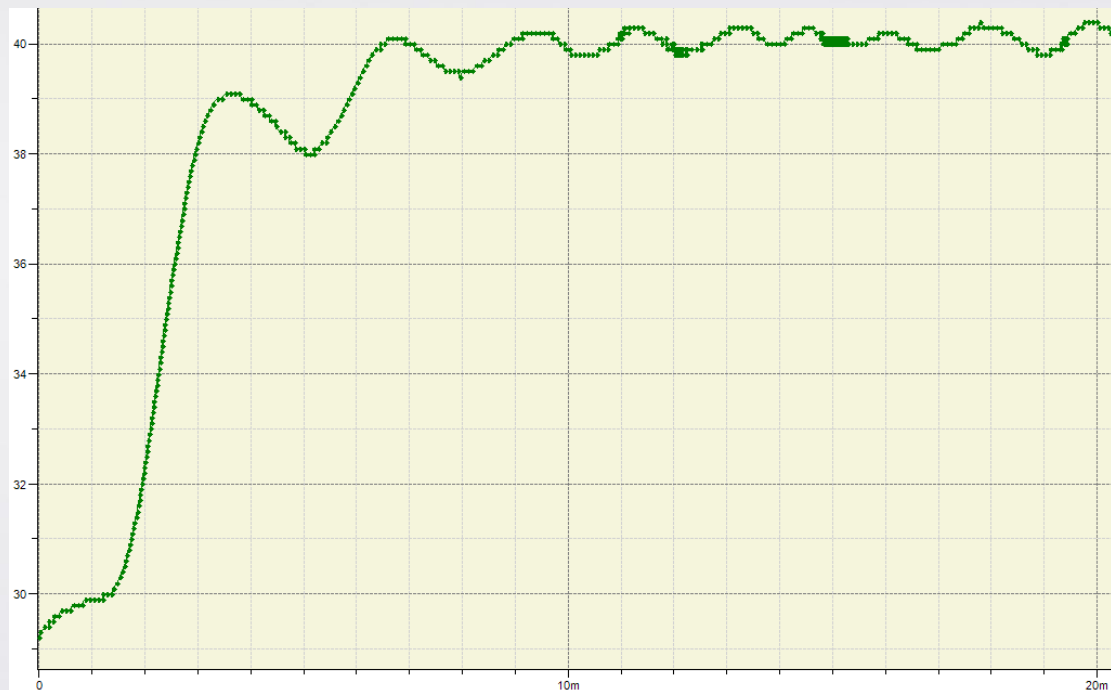
等待溫度下降低於 SV 再啟動 Full power

等待溫度再次下降低於 SV 代表 Auto Tune 完成

此時，自動計算 K_p 、 K_i 、 K_d 、 T_f 輸出至標籤

Auto Tune 原始控制曲線

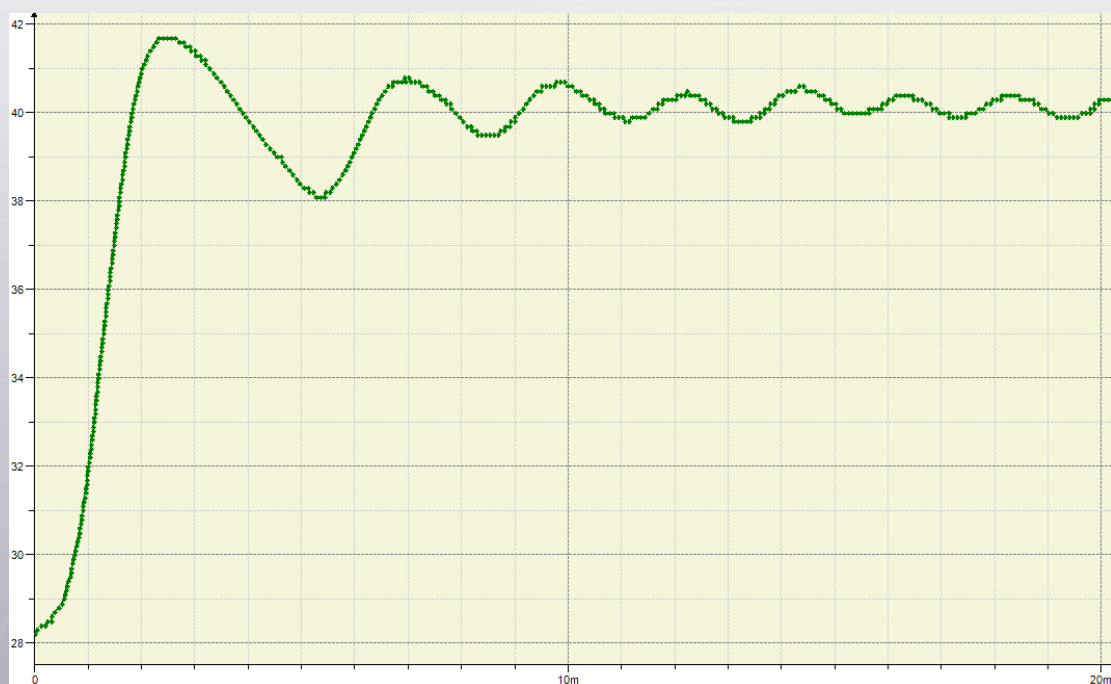
再利用這組參數重新加熱的溫度曲線如下：



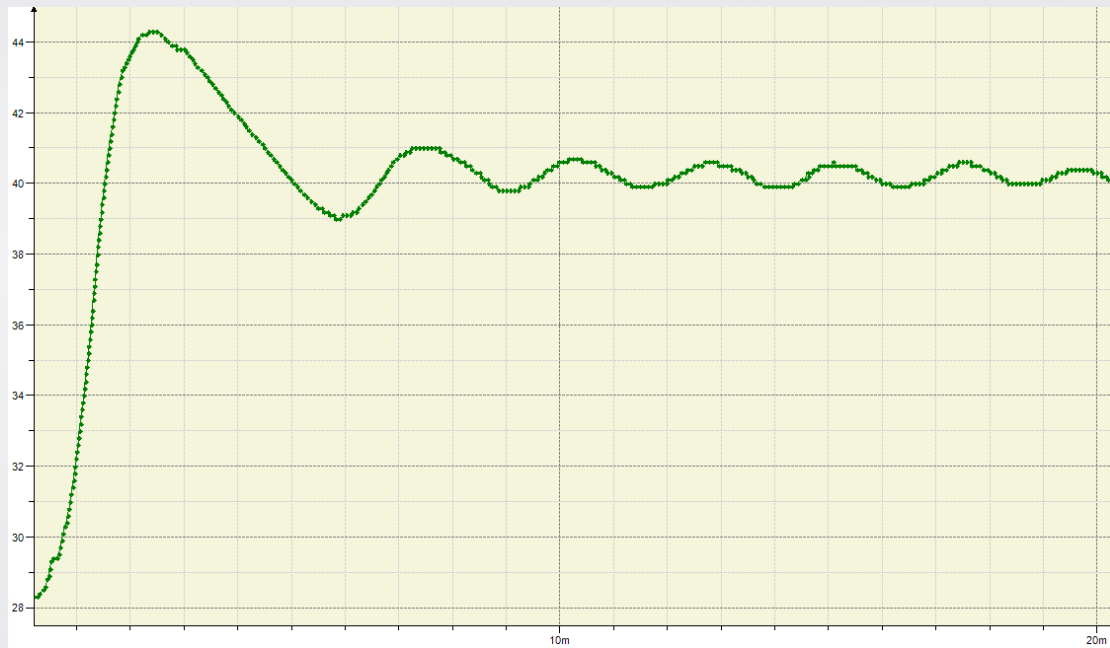
最大超越量低於目標值 SV，故手動調整參數。

Auto Tune 手動調整控制曲線

降低 K_p 參數再重新加熱，曲線如下：



根據上圖可見，最大超越量已經大於目標值 SV，但收斂時間較長（大於 10 分鐘），此時調整 Ki 參數重新加熱，曲線如下：



調整後溫度曲線