

**Instruction Manual**

# SYL-2342P Ramp and Soak PID Temperature Controller

Version 1.6 (Nov, 2022)

**Caution**

- This controller is intended to control equipment under normal operating conditions. Failure or malfunction of the controller may lead to abnormal operating conditions, which result in personal injury or damage to the equipment or other property. Devices (limit or safety controls) or systems (alarm or supervisory) intended to warn of or protect against failure or malfunction of the controller must be incorporated into and maintained as part of the control system.
- Installing the rubber gasket supplied will protect the controller front panel from dust and water splash (IP54 rating). Additional protection is needed for higher IP rating.
- This controller carries a 90-day warranty. This warranty is limited to the controller only.

**1. Features**

- 30 programmable steps for ramp/soak process control.
- High flexibility in program and operation. It has programmable/maneuverable commands such as jump (for loops), run, hold and stop. The program can even be modified while it is running.
- The program can also control the two relays that are used for alarms. This feature can be used to notify the operator of the stage of the operation, or to signal other equipment.
- The safety-start and ready function may allow the program to run more efficiently. Six power-off/power-on event handling (see 5.3.15) modes are available, which can prevent the program control from being adversely affected by unexpected power interruptions.

**2. Specifications**

Table 1. SYL-2342P specifications.

<b>Input type</b>	Thermocouple (TC): K, E, S, N, J, T, B, WRe5/ 26; RTD (Resistance Temperature Detector): Pt100, Cu50 DC Voltage: 0~5V, 1~5V, 0~1V, -100~100mV, -20 ~ 20mV, -5~5V, 0.2~1V DC current: 0~10mA, 1~10mA, 4~20mA. (Use external shunt resistor for higher current)
<b>Input range</b>	Please see section 5.3.6 for detail.
<b>Accuracy</b>	± 0.2% Full scale: RTD, linear voltage, linear current and thermocouple input with ice point compensation or Cu50 copper compensation. 0.2% Full scale or ± 2 °C: Thermocouple input with internal automatic compensation. Note: For thermocouple B, the measurement accuracy of ± 0.2% can only be guaranteed when input range is between 600~1800 °C.
<b>Response time</b>	≤ 0.5s (when FILt = 0)
<b>Display resolution</b>	1°C, 1°F; or 0.1°C
<b>Control mode</b>	Fuzzy logic enhanced PID control On-off control
<b>Output mode</b>	Relay contact (NO): 250VAC/7A, 120V/10A, 24VDC/10A
<b>Alarm relay rating</b>	Relay contact (NO): 250VAC/1A, 120VAC/3A, 24V/3A
<b>Alarm function</b>	Process high alarm, process low alarm, deviation high alarm, and deviation low alarm

<b>Power supply</b>	85~260VAC/50~60Hz
<b>Power consumption</b>	≤ 5 Watt
<b>Ambient temperature</b>	0~50°C, 32~122°F
<b>Dimension</b>	48 x 48 x 100mm (W x H x D)
<b>Mounting cutout</b>	45 x 45mm

**3. Terminal Wiring**

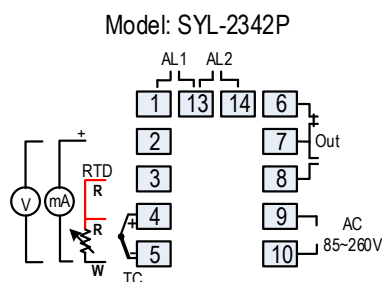


Figure 1. Wiring terminals of SYL-2342P.

**3.1. Sensor connection**

Please refer to Table 3 for the input sensor type (Sn) setting codes. The initial setting for input is for a K type thermocouple. Set Sn to the right sensor code if another sensor type is used.

**3.1.1. Thermocouple**

The thermocouple should be connected to terminals 4 and 5. Make sure that the polarity is correct. There are two commonly used color codes for the K type thermocouple. US color code uses yellow (positive) and red (negative). Imported DIN color code uses red (positive) and green/blue (negative). The temperature reading will decrease as temperature increases if the connection is reversed.

When using ungrounded thermocouple that is in touch with a large conductive subject, the electromagnetic field picked up by the sensor tip might be too large for the controller to handle, i.e., the temperature display will change erratically. In that case, connecting the shield of thermocouple to terminal 5 (circuit ground of the controller) might solve the problem. Another option is to connect the conductive subject to terminal 5.

**3.1.2. RTD sensor**

For a three-wire RTD with standard DIN color code, the two red wires should be connected to the terminals 3 and 4, and the white wire should be connected to terminal 5. For a two-wire RTD, the wires should be connected to terminals 4 and 5. Jump a wire between terminals 3 and 4. Set controller input type Sn to 21.

**3.1.3. Linear input (V, mV, mA or resistance)**

Voltage (V) and milli-ampere (mA) current signal inputs should be connected between terminals 2 and 5. Terminal 2 is positive. Milli-voltage (mV) signal inputs should be connected between terminals 4 and 5. Terminal 4 is positive. For resistance inputs, short terminals 3 and 4, then connect resistance inputs between terminals 4 and 5.

**3.2. Power to the controller**

The power cables should be connected to terminals 9 and 10. Polarity does not matter. It can be powered by 85-260V AC power source. Neither a transformer nor jumper is needed to wire it up. For the sake of consistency with the wiring example described later, we suggest you connect the hot wire to terminal 9 and neutral to 10.

**3.3. Control output connection**

The relay output of the controller SYL-2342P can be used to turn on a contactor or a solenoid valve. It can drive a small heater directly if that draws less than 10A at 120V AC power source. For applications needing two control outputs, such as one for heating and another for cooling, relays AL1 or AL2 can be used for the second output with on/off control mode. Please see Figure 14 for details.

**3.4. Connecting the load through a contactor**

Assuming the controller is powered by 120V AC and the contactor has a 120V AC coil, jump a wire between terminals 8 and 9. Connect terminal 7 to one lead of the coil and terminal 10 to the other lead of the coil. Please see Figure 7 for example.

**3.5. Connecting the load directly to the internal relay**

Assuming the controller and the load (heater or cooler) are powered by the same voltage. Jump a wire from terminal 9 to 8. Connect terminal 7 to the one lead of the load and terminal 10 to the other lead of the load. Please see Figure 6 and 9 for details.

**3.6. For first time users without prior experience with PID controllers, the following notes may prevent you from making common mistakes.**

Power to the heater does not flow through terminal 9 and 10 of the controller. The controller consumes less than 2 watts of power. It only provides a control signal to the relay. Therefore, wires in the 18 to 26 gauge range should be used for providing power to terminals 9 and 10. Thicker wires may be more difficult to install.

The relay outputs, AL1 and AL2, are “dry” single pole switches. They do not provide power by themselves. Please see Figure 7 and 9 for how they are wired when 120V power is supplied. The wiring diagram in Figure 9 can also be applied to situations where both the load and the controller are using 240V power supply. If the load of the relay requires a different voltage than that for the controller, another power source will be needed. See Figure 14 for examples.

The power is controlled by regulating the duration of on time for a fixed period of time rather than the amplitude of the voltage or current. This is often referred as time proportional control. For example, if the cycle rate is set for 100 seconds, a 60% output means controller will switch on the power for 60 seconds and off for 40 seconds (60/100 = 60%). Almost all high power control systems use time proportional control, because amplitude proportional control is too expensive and inefficient.

**4. Front Panel and Display Mode**

**4.1. Front Panel**

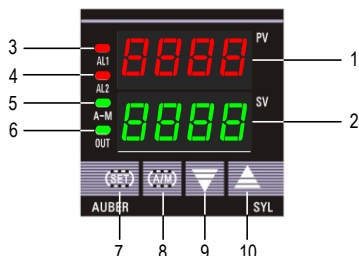


Figure 2. Front panel.

- ① PV window: Displays the sensor read out, or process value (PV).
- ② SV window: Displays the set value (SV) or output value (%).
- ③ AL1 indicator: It lights up when AL1 relay is on.
- ④ AL2 indicator: It lights up when AL2 relay is on.
- ⑤ A-M indicator: The light indicates the controller operating status. When A-M is solid on, program is running. When A-M is flashing, program is paused. When A-M is off, program is stopped.
- ⑥ Output indicator: It is synchronized with control output (terminal 7 and 8), and the power to the load. When it is on, the heater (or cooler) is powered.
- ⑦ SET key: In normal operating mode, keep pressing the SET key momentarily so the display will be switched between the temperature, step number and step timer. Press and hold the SET key for 2s so the controller will switch to the parameter setting mode. For details, see Figure 3 below.
- ⑧ A/M key: Programming key; digit shift key.
- ⑨ DOWN key ▼: Decreases numeric value of the setting value.
- ⑩ UP key ▲: Increases numeric value of the setting value.

**4.2. Controller display mode**

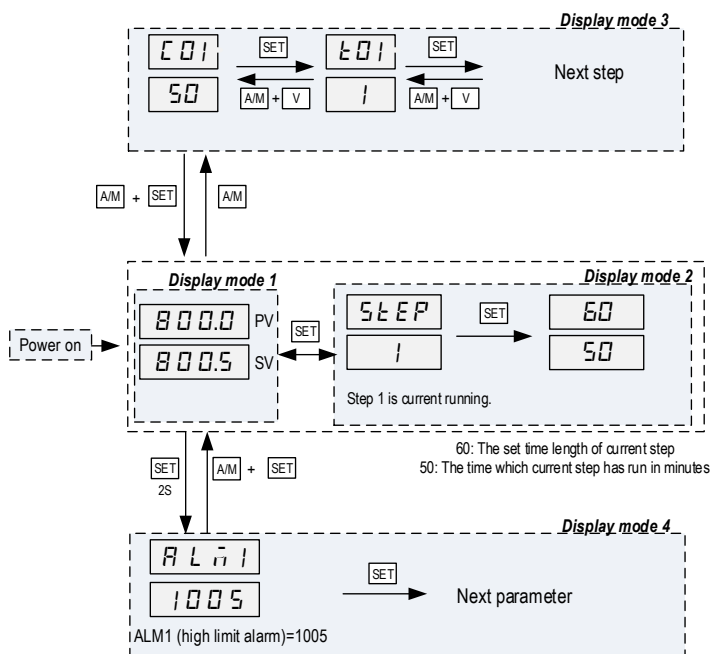


Figure 3. Display modes.

**Display mode 1: normal operating mode.** When the power is turned on, the upper display window shows the measured value (PV) and the lower window shows the four-digit set value (SV).

**Display mode 2: checking the step.** Press the SET key once to change the display from mode 1 to mode 2. The upper display window shows “STEP” and the lower windows shows the current step number. Press SET key again to show the timer information. PV window shows the set time for the current step. SV window shows remaining time in minutes. Press SET key again to return to the display mode 1.

**Display mode 3: programming mode.** Press A/M key once to change the display from mode 1 to mode 3. This mode is used for setting and changing the program.

**Display mode 4: Parameter setting mode.** Press and hold the SET key for 2 seconds to enter the display mode 4. The top window shows the name of a parameter and the bottom window shows its value. Use the UP and DOWN arrow key the change the value; use the SET key to save the change and go to the next parameter.

5. Parameter Settings

5.1. Parameter setup mode

When the display mode is 1 or 2, press SET and hold for roughly 2 seconds until the parameter setup menu is displayed (display mode 3). Please refer to 4.3 for how to set the parameters.

5.2. Setup flow chart

While in the parameter setup mode, use ▲ and ▼ to modify a digit and use A/M and SET key at the same time to exit the parameter setup mode; otherwise, the instrument will automatically exit if no key is pressed for 10 seconds. Figure 4 is the setup flow chart. Please note that changed parameter will be automatically registered without pressing the SET key. If the controller is locked (see 7.16), only limited parameters (or no parameters) can be changed.

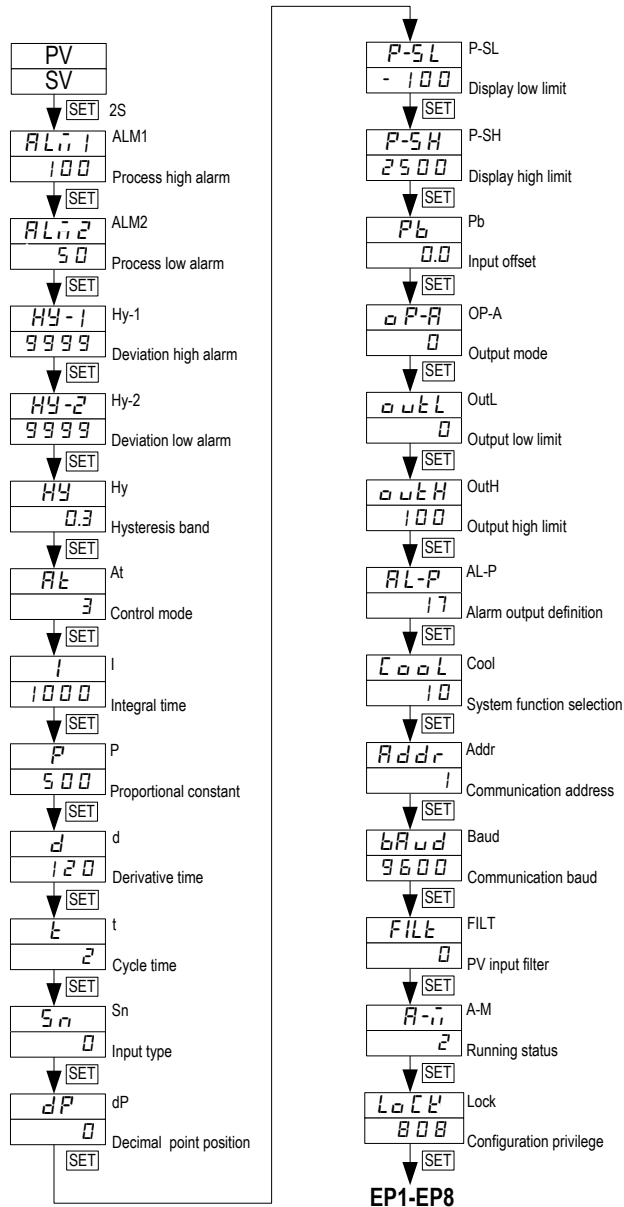


Figure 4. System setup flow chart.

5.3. List of system parameters

Table 2. System parameters.

Code	Description	Setting Range	Initial Setting	Remarks
ALM1	Process high alarm	-1999~+9999°C or °F	100	See 5.3.1
ALM2	Process low alarm	-1999~+9999°C or °F	50	
Hy-1	Deviation high alarm	0~+9999°C or °F	9999	
Hy-2	Deviation low alarm	0~+9999°C or °F	9999	
Hy	Hysteresis Band	0~200°C or °F or 0~2000 for linear input	0.3	See 5.3.2
At	Auto tuning	0~3. Set to 1 or 2 to start Auto tuning	3	See 5.3.3
I	Integral time	0~9999	1000	See 5.3.4
P	Proportional Constant	1~9999%	500	
d	Derivative Time	0~2000	120	
t	Cycle time	2~125	20 for relays	See 5.3.5
Sn	Input type	0~37	0 (K type TC)	See 5.3.6
dP	Decimal point position	0~3	0	See 5.3.7
P-SL	Display low limit	-1999~+9999°C or °F	-100	See 5.3.8
P-SH	Display high limit	-1999~+9999°C or °F	2500	
Pb	Input offset	-1999~+4000 -1999~+9999°C or °F	0.0	See 5.3.9
OP-A	Output mode	0~2	0	See 5.3.10
OUTL	Output low limit	0~100%	0	See 5.3.11
OUTH	Output high limit	0~100%	100	
AL-P	Alarm output definition	0~31	17	See 5.3.12
COOL	System function selection	0~15	10	For heating and °F display, see 5.3.13
Addr	Communication address	s 0~20	1	Ignore this setting
bAud	Communication baud rate	0~19200	9600	Ignore this setting
FILT	PV input filter	0~20	0	See 5.3.14
A-M	Control Mode Parameter	0~6	2	See 5.3.15
Lock	Configuration privilege	0~9999	808	All parameters are unlocked. See 5.3.16
EP1-EP8	Field parameter definition	nonE ~ A-M	nonE	To be defined by user. See 5.3.16

5.3.1. Alarm parameters

This controller offers four types of alarm, “ALM1”, “ALM2”, “Hy-1”, “Hy-2”.

- ALM1: High limit absolute alarm. If the process value is greater than the value specified as “ALM1 + Hy” (Hy is the Hysteresis Band), then the alarm will turn on. It will turn off when the process value is less than “ALM1 - Hy”.
- ALM2: Low limit absolute alarm. If the process value is less than the value specified as “ALM2 - Hy”, then the alarm will turn on, and the alarm will turn off if the process value is greater than “ALM2 + Hy”.
- Hy-1: Deviation high alarm. If the temperature is above “SV + Hy-1 + Hy”, the alarm will turn on, and the alarm will turn off if the process value is less than “SV + Hy-1 - Hy” (we will discuss the role of Hy in the next section)
- Hy-2: Deviation low alarm. If the temperature is below “SV - Hy-2 - Hy”, the alarm will turn on, and the alarm will turn off if the temperature is greater than “SV - Hy-2 + Hy”.

**Things you should know about alarms**

**1) Absolute alarm and deviation alarm**

High (or low) limit absolute alarm is set by the specific temperatures that the alarm will be triggered. Deviation high (or low) alarm is set by how many degrees above (or below) the target temperature (SV) that the alarm will be on.

Assuming ALM1 = 1000 °F, Hy-1 = 5 °F, Hy = 1, SV = 700 °F. When the probe temperature (PV) is above 706 °F, the deviation alarm will be on. When the temperature is above 1001 °F, the process high alarm will be on. Later, when SV changes to 600 °F, the deviation alarm will be changed to 606 °F but process high alarm will remain the same.

**2) Assignment of the relays for the alarms**

AL1 and AL2 are the names of the two relays used for alarm output. AL1 is the alarm relay 1 (terminal 1 and 13) and AL2 is alarm relay 2 (terminal 13 and 14). Please do not confuse the relays with alarm parameter ALM1 (process high alarm) and ALM2 (process low alarm). The parameter AL-P (alarm output definition) allows user to select which relay to trigger when the alarm set condition is met. You can set multiple alarms to activate one relay (either AL1 or AL2), but you can't activate both relays with just one alarm. Please also note that deviation alarms (Hy-1 and Hy-2) cannot trigger alarm relay AL1.

**3) Display of the alarm**

When AL1 or AL2 relay is activated, the LED on the upper left will light up. If you have multiple alarms assigned to a single relay, you might want to know which alarm activated the relay. This can be done by setting the E constant in formula of calculating the AL-P (see section 5.3.12). When E = 0, the bottom display of the controller will alternately display the SV and the activated alarm type.

**4) Activate the AL1 and AL2 by program command**

Relay AL1 and AL2 can be programed to pull in or drop out at specified program steps. This is discussed in the section 6.2.9 Commands for Special Event.

**5.3.2. Hysteresis band "Hy"**

The Hysteresis Band parameter Hy is also referred as Dead Band or Differential. It permits protection of the on/off control from high switching frequency caused by process input fluctuation. Hysteresis Band parameter is used for on/off control, 4-alarm control, as well as the on/off control at auto tuning. For example: (1) When controller is set for on/off heating control mode, the output will turn off when temperature goes above SV + Hy and on again when it drops to below SV - Hy. (2) If the high alarm is set at 800 °F and hysteresis is set for 2 °F, the high alarm will be on at 802 °F (ALM1 + Hy) and off at 798 °F (ALM1 - Hy).

Please note that the cycle time can also affect the action. If the temperature passes the Hy set point right after the start of a cycle, the controller will not respond to the Hy set point until the next cycle. If cycle time is set to 20 seconds, the action can be delay as long as 20 seconds. Users can reduce the cycle time to avoid the delay.

**5.3.3. Control mode "At"**

**At = 0:** on/off control mode. It works like a mechanical thermostat. It is suitable for devices that do not like to be switched at high frequency, such as motor and valves. See section 5.3.4, section 2) for details.

**At = 1:** get the controller ready to start the Auto tuning process by pressing and holding the A/M key for 2 seconds.

**At = 2:** start auto tuning. It will start automatically after 10 seconds. The function is the same as starting auto tuning from front panel.

**At = 3:** controller works in PID mode. This configuration is automatically set after auto tuning is done. Auto tuning from the front panel is inhibited to prevent accidental re-starting of the auto tuning process. To start auto tuning again, set At = 1 or At = 2.

For more details on each control mode, please read the next section 5.3.4.

**5.3.4. Control mode**

**1) PID control mode**

Please note that because this controller uses fuzzy logic enhanced PID control algorithm, the definition of the control constants (P, I and d) are different than that of the traditional proportional, integral, and derivative parameters.

In most cases the fuzzy logic enhanced PID control is very adaptive and may work well without changing the initial PID parameters. If not, users may need to use auto-tune function to let the controller determine the parameters automatically. If the auto tuning results are not satisfactory, you can manually fine-tune the PID constants for improved performance. Or you can try to modify the initial PID values and perform auto tune again. Sometimes the controller will get the better parameters.

**(1) Proportional constant "P"**

Please note that the P constant is not defined as Proportional Band as in the traditional model. Its unit is not in degrees. A larger constant results in larger and quicker action, which is the opposite of the traditional proportional band value. It also functions in the entire control range rather than a limited band.

If you are controlling a very fast response system (> 1°F/second) that fuzzy logic is not quick enough to adjust, set P = 1 will change the controller to the traditional PID system with a moderate gain for the P.

**(2) Integral time "I"**

Integral action is used to eliminate offset. Larger values lead to slower action. Increase the integral time when temperature fluctuates regularly (system oscillating). Decrease it if the controller is taking too long to eliminate the temperature offset. When I = 0, the system becomes a PD controller.

**(3) Derivative time "D"**

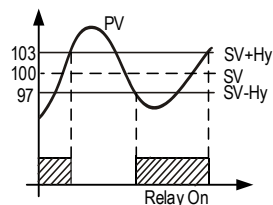
Derivative action can be used to minimize the temperature overshoot by responding to its rate of change. The larger the number, the faster the action.

**2) On/off control mode**

It is necessary for inductive loads such as motors, compressors, or solenoid valves that do not like to take pulsed power. It works like a mechanical thermostat. When the temperature difference between PV and SV passes hysteresis band (Hy), the heater (or cooler) will be turned off. When the temperature difference drops back to below the hysteresis band, the heater (or cooler) will turn on again.

To use the on/off mode, set At = 0. Then, set Hy to the desired range based on control precision requirements. Smaller Hy values result in tighter temperature control, but also cause the on/off action to occur more frequently.

Note: Ramp & soak program is still in use under ON/OFF mode.



When heating, At=0  
 If  $PV \leq (SV - Hy)$ , relay on  
 If  $PV \geq (SV + Hy)$ , relay off  
 (SV=100, Hy=3)

Figure 5. On/off control mode.

**3) Auto-tune**

The auto-tune can be started in two ways: 1) Set At = 1, then you can start the auto-tune any time during the normal operation by holding the A/M key for 2 seconds (short press the A/M key will bring the controller to the programming mode). 2) Set At = 2. It will start automatically after 10 seconds; In this ramp/soak controller, user can start the auto-tune any time when a program is running, on hold, or stopped. When auto-tuning starts, the lower window will flash between "At" and the current set temperature.

During auto tuning, the instrument executes on/off control at the current set temperature. After 2-3 times on/off action, the microprocessor in the instrument will analyze the period, amplitude, waveform of the oscillation generated by the on/off control, and calculate the optimal value for the P, I, and D.

When auto-tune is finished, the lower window will stop flashing "At"; only the set temperature will be shown in the lower window. Controller will resume running the program. Generally, you will only need to perform auto tuning once. After the auto tuning is finished, controller will set parameter "At" to 3, which will prevent the (A/M) key from triggering auto-tune. This will prevent an accidental repeat of the auto-tuning process.

The auto-tuning process can also be manually stopped by holding the A/M key 2 seconds. Hold the A/M key too long may bring the controller into the programming mode. If the auto-tune is manually stopped, controller will not change "At" to 3. P, I, and D will not be changed either. User can start the auto-tune again by holding A/M key for 2 seconds.

For this controller, the auto-tune should not be activated when the controller is ramping because the result obtained at one temperature is optimized for that temperature only. The auto-tune should be activated at the most critical temperature where you need to hold the temperature in tight precision. User should program a long soak step for the critical temperature and only activate the auto-tune after the process temperature has reached the set temperature.

For example, if 500°F is the most important temperature you need to hold in tight, set C01 = 80, T01 = 30, C02 = 500, T02 = 300, C03 = 500, T03 = -121. When temperature reaches 500°F, set AT = 2 to activate the auto-tune. When auto-tuning is finished, set the steps back to the original program. The PID parameters obtained at 500°F may work fine at temperatures higher than 500°F (but it may be a little slow). But these parameters may not work well at lower temperature range.

**5.3.5. Cycle time "t"**

It is the time period (in seconds) that the controller uses to calculate its output. For example, when t = 20, if the controller decides output should be 10%, the heater will be on 2 second and off 18 seconds for every 20 seconds. For relay or contactor output, it should be set longer to prevent contacts from wearing out too soon. Normally it is set to 20~40 seconds.

**5.3.6. Input selection code for "Sn"**

Table 3. Code for Sn and its range.

Sn	Input device	Display range (°C)	Display range (°F)	Wiring Pins
0	K (thermocouple)	-50~+1300	-58~2372	4, 5
1	S (thermocouple)	-50~+1700	-58~3092	4, 5
2	WRe (5/26)(thermocouple)	0~2300	32~4172	4, 5
3	T (thermocouple)	-200~350	-328~662	4, 5
4	E (thermocouple)	0~800	32~1472	4, 5
5	J (thermocouple)	0~1000	32~1832	4, 5
6	B (thermocouple)	0~1800	32~3272	4, 5
7	N (thermocouple)	0~1300	32~2372	4, 5
20	Cu50 (RTD)	-50~+150	-58~302	3, 4, 5
21	Pt100 (RTD)	-200~+600	-328~1112	3, 4, 5
26	0 ~ 80 Ω	-1999~+9999 Defined by user with P-SL and P-SH		3, 4, 5
27	0 ~ 400 Ω		3, 4, 5	
28	0 ~ 20 mV		4, 5	
29	0 ~ 100 mV		4, 5	
30	0 ~ 60 mV		4, 5	
31	0 ~ 1000 mV		4, 5	
32	200 ~ 1000 mV, 4-20 mA (w/ 50Ω Resistor)		4, 5	
33	1 ~ 5 V 4~20 mA (w/ 250Ω Resistor)		2, 5	
34	0 ~ 5 V		2, 5	
35	-20 ~ +20 mV		4, 5	
36	-100 ~ +100 mV		4, 5	
37	-5 ~ +5V		2, 5	

Please see Table 3 for the acceptable sensor type and its range.

**5.3.7. Decimal point setting "dP"**

The parameter "dP" defines how many digits after the decimal point will be displayed, please see Table 4 for its value the corresponding resolution.

**1) Thermocouple and RTD**

For thermocouples and RTD sensors, dP can only be set to 0 or 1. When dP = 0, temperature display resolution is 1 °C (°F). When dP = 1, temperature display resolution is 0.1°C. The 0.1 degree resolution is only available for Celsius display. The temperature will be displayed at the resolution of 0.1 °C for input below 1000 °C and 1 °C for input over 1000 °C.

**2) Linear input (voltage, current, or resistance input, Sn = 26 ~ 37)**

For other linear input signals, dP can be set to from 0 to 3.

Table 4. dP parameter setting.

dP Value	0	1	2	3
Display format	0000	000.0	00.00	0.000

**5.3.8. Limiting the control range, "P-SH" and "P-SL"**

**1) Thermocouple and RTD**

For temperature sensor input, the "P-SH" and "P-SL" values define the set value range. P-SL is the low limit, and P-SH is the high limit. For example, sometimes you may want to limit the temperature setting range so that the operator can't set a very high temperature by accident. If you set the P-SL = 100 and P-SH = 130, operator will only be able to set the temperature between 100 and 130.

**2) Linear input (voltage, current, or resistance input, Sn = 26 ~ 37)**

For linear input signals, "P-SH" and "P-SL" are used for scaling the display. "P-SL" is the value to be displayed when the signal is at its low limit of the linear input. "P-SH" is the value to be displayed when the signal is at its high limit of the linear input. For example, for 0-5V signal, "P-SL" corresponds to the value when signal is 0V, and "P-SH" corresponds to the value when signal is 5V. If you have a pressure transducer that its 0V output represents -18 PSI, and 5V output represents 168 psi, then set P-SL = -18 and P-SH = 168 will convert the voltage signal to pressure value in PSI and display it on the PV window on the controller. In the real world, some of the pressure transducer manufactures only offer the signal in 0.5 V to 4.5 V range. In that case, user needs to extrapolate the signal range to get the corresponding value at 0 V and 5 V assuming it is linear. .

**5.3.9. Input offset "Pb"**

Input offset Pb is used to add an offset value to compensate the sensor error or simply to shift the reading. For example, if the controller displays 2 °C when probe is in ice/water mixture, setting Pb = -2, will make the shift the temperature reading to 0 °C.

**5.3.10. Output definition "OP-A"**

This parameter is not used for this model. It should not be changed.

**5.3.11. Output range limits "OUTL" and "OUTH"**

OUTL and OUTH allow you set the output range low and high limit. OUTL is a useful feature for a system that needs to have a minimum amount of power as long as the controller is powered. For example, if OUTL = 20, the controller will maintain a minimum of 20% power output even when input sensor failed.

OUTH can be used when you have an overpowered heater to control a small subject. For example, if you set the OUTH = 50, the 5000 watt heater will be used as 2500W heater (50%) even when the PID wants to send 100% output.

### 5.3.12. Alarm output definition "AL-P"

Parameter "AL-P" is used to determine which alarms ("ALM1", "ALM2", "Hy-1" and "Hy-2") is assigned to which relay (AL1 or AL2). It can be configured to a value in the range of 0 to 31. But **please note** that the deviation alarms, Hy-1 and Hy-2, cannot be assigned to relay AL1. Its function is determined by the following equation:

$$AL-P = AX1 + BX2 + CX4 + DX8 + EX16,$$

If A = 0, then AL2 is activated when Process High Alarm occurs.  
 If A = 1, then AL1 is activated when Process High Alarm occurs.  
 If B = 0, then AL2 is activated when Process Low Alarm occurs.  
 If B = 1, then AL1 is activated when Process Low Alarm occurs.  
 If C = 0, then AL2 is activated when Deviation High Alarm occurs.  
 If C = 1 (**not available on this controller**), then AL1 is activated when Deviation High Alarm occurs.  
 If D = 0, then AL2 is activated when Deviation Low Alarm occurs.  
 If D = 1 (**not available on this controller**), then AL1 is activated when Deviation Low Alarm occurs.  
 If E = 0, then alarm types, such as "ALM1" and "Hy-2" will be displayed alternatively in the lower display window when the alarms are on. This makes it easier to determine which alarms are on.  
 If E = 1, the alarm will not be displayed in the lower display window (except for "orAL"). Generally this setting is used when the alarm output is used for control purposes.

For example, in order to activate AL1 when a Process High Alarm occurs, trigger AL2 by a Process low alarm, Deviation high alarm, or Deviation low alarm, and not show the alarm type in the lower display window, set A = 1, B = 0, C = 0, D = 0, and E = 1. Parameter "AL-P" should be configured to:

$$AL-P = 1X1 + 0X2 + 0X4 + 0X8 + 1X16 = 17$$

#### (AL-P = 17 is the factory default setting)

Unlike other controllers that have only one type of alarms (either absolute or deviation alarm), this controller has both and allows both types to function simultaneously. If you only want to use one type of alarm, set the other alarm parameters to maximum or minimum (i.e., set ALM1, Hy-1, and Hy-2 to 9999, ALM2 to -1999) to stop its function.

### 5.3.13. "COOL" for Celsius, Fahrenheit, Heating, and Cooling Selection

Parameter "COOL" is used to set the display unit, heating or cooling, and alarm suppression. Its value is determined by the following formula:

$$COOL = AX1 + BX2 + CX8,$$

A = 0, reverse action control mode for heating control.  
 A = 1, direct action control mode for cooling control.  
 B = 0, alarm suppressing setting, **not available on this model of controller**.  
 B = 1, alarm suppressing setting, **not available on this model of controller**.  
 C = 0, display unit in °C.  
 C = 1, display unit in °F.  
 The factory setting is A = 0, B = 1, C = 1 (heating mode, display temperature in Fahrenheit). Therefore,

$$COOL = 0X1 + 1X2 + 1X8 = 10.$$

To change from Fahrenheit to Celsius display, set COOL = 2.

### 5.3.14. Input digital filter "FILt"

If measurement input fluctuates due to noise, then a digital filter can be used to smooth the input. "FILt" may be configured in the range of 0 to 20. Stronger filtering increases the stability of the readout display, but causes more delay in the response to change in temperature. FILt = 0 disables the filter.

### 5.3.15. Control Mode Parameter "A-M"

The function of the A-M parameter is defined differently in the SYL-23X2P than it is for the controller without the ramp/soak option (SYL-23X2). Its operation is determined according to the equation:

$$A-M = AX1 + BX4,$$

where "A" is used to select one of the three power outage/startup event handling modes, and "B" is used to select one of two hold modes.

#### Power Outage/Startup Modes:

**A = 0:** When the instrument is turned on, the program will simultaneously jump to 29<sup>th</sup> program segment and clear event output status. This mode is suitable for applications in which power failure is not allowed at any time. The user may do error handling in segment 29, such as switching on the event output to trigger an alarm.

**A = 1:** If there is no deviation alarm at power up, the program will continue running from the original break point and the event output state remains. Otherwise, the program will jump to the 29<sup>th</sup> segment and clear event output status.

**A = 2 (default):** After power is turned on, it will continue the program from the original break point, and the event output state will remain. This mode is suitable for the applications in which power failure does not affect production.

#### Hold Mode definition:

**B = 0 (default):** When the controller is put into "hold" mode, the PV is maintained at the current SV.

**B = 1:** the output of the controller is at OUTL when it is on hold.

The default setting of this controller is A-M = 2 (A = 2 and B = 0).

### 5.3.16. Configuration Privilege "Lock" and Field Parameters "EP"

To prevent the parameters and the program being changed accidentally, you can completely or partially lock the parameters and the program after the initial setup. The configuration privilege is determined by "lock". Please refer to the Table 5 for the privilege levels.

#### 1) "Lock" value

"Lock" value determines the privilege level. It can be set to any value between 0 and 9999, but only certain values are defined. The factory default setting for lock is 808, all parameters are unlocked, which allows all parameters and program steps being viewed and edited during operation. When Lock value is set to 0, 1, or 2, please refer to Table 5 for the corresponding privilege value. When Lock is set to 3 or any other value above (except 808), only the field parameters can be viewed but not editable.

**Note:** In any situation that parameters that are locked, you can always view and change the "lock" value after you view all field parameters. Change the value will immediately change the privilege level.

#### 2) Field Parameters EP1 ~ EP8

By assigning system parameters as Field Parameters (EP1 ~ EP8), you can select which parameter can be displayed or changed when controller is locked. Up to 8 parameters can be assigned as Field Parameter. The Field Parameter can be any parameter listed in Table 2 except Field Parameters themselves and the Lock parameter.

When Lock is set to 0, 1, 2, and so on, only parameters defined in an EP can be displayed. This function can speed up parameter modification and prevent critical parameters (like input, output parameters) from being modified. If the number of field parameters is less than 8, then define the first unused parameter as none. For example, if only ALM1 and ALM2 need to be modified by field operators, the parameter EP can be set as following:

Lock = 2, EP1 = ALM1, EP2 = ALM2, EP3 = nonE.

In this case, the controller will ignore the field parameters from EP4 to EP8. If field parameters are not needed after the instrument is initially adjusted, simply set EP1 to nonE.

Lock code 0, 1 and 2, will give the operator limited privileges to change some of the parameters that can be viewed. Table 5 shows the privileges associated with each lock code.

**3) Program Adjustment**

When Lock = 0, program adjustment is allowed. The operator can view the program steps and change the setting. When Lock = 1, 2, 3 and up, program steps cannot be viewed nor edited.

**4) Step Selection**

When Lock = 0 or 1, operator can check the current step number and jump to another step by changing the step number. When Lock = 2, 3 or up, operator is not allowed to check the current step number and cannot jump to other steps.

Table 5. "Lock" value and the configuration privilege level.

Lock value	Privilege	EP1 - 8 Adjustment	Program Adjustment	Step Selection
0	limited	Yes	Yes	Yes
1		Yes	No	Yes
2		Yes	No	No
3 and up		No	No	No
808 (Default)	unlimited	Yes	Yes	Yes

Note: to limit the control temperature range instead of completely locking it, please refer to section 5.3.8.

**6. Program Ramp/Soak on the Controller**

**6.1. Terminology**

**Program step (SLEP):** is a control step with its set value and set time are specified. The Step Number (n) can range from 1 to 30. The Current Step is the step that is being executed.

**Step n:** is the n-th step in the program. Its set temperature is expressed as C[n]; its set time is expressed as t[n]. For example, for the 1<sup>st</sup> step, the set temperature is C01 (E □ t), the set time is t01 (E □ t).

**Step set temperature (C[n]):** is the set point at the beginning of step n.

**Step set time (t[n]):** is the time from step n to the next step (n + 1). The unit is in minutes and the valid range is from 1 to 9999.

**Running time:** Running time is how long the current step has been running. When the running time reaches the step set time, the program will jump to the next step automatically.

**Jump:** Go to a specified program step. This step can be any steps in the program (step number 1 to 30). This feature can be used to perform a control loop or to skip certain steps. While checking the current step number, if the step number is modified, the program will also jump. Furthermore, once the controller reached and finished the 30<sup>th</sup> step, it will jump back to the first step and continue running unless it is specified other ways.

**Run (run):** The program is being executed. When the program is under the "run" status, the "A/M" indicator is solid on, the timer counts down, and the set point value changes according to the preset ramp curve.

**Hold (Hold):** The execution of the program is paused. When the program is on "hold", the "A/M" indicator should be blinking, the temperature is still being controlled, but the timer is paused so the current set point remains. Under the "hold" status, the output from the controller can be defined by parameter "A-M" (see section 5.3.15).

**Stop (Stop):** Execution of the program is stopped. When the program is stopped, the lower window will flash between "stop" and the current set value, the timer and the output control will stop, and the running time and event output switch will reset. If the "run" operation is activated while the controller is in the "stop" status, the program will start-up and run from the step 1.

**Power interrupt:** It means the power has turned off or an unexpected power failure has occurred during running status. Six handling modes are available to the user. Please see 5.3.15 for details. Manually powering off the controller while the program is running is also considered as an occasion of power interrupt.

**Event output:** A programmed relay action. The two alarm relays (AL1 and AL2) can be programmed to pull in or drop out to activate or deactivate external equipment.

**Safety-start:** If the difference between the PV and SV is larger than the deviation alarm settings at the beginning of a step (or at powering up), controller will put program to "hold", "A/M" light will be blinking, and the timer won't start running until PV falls within the safety range (see "safety range" below). During this time, controller will try to bring the temperature to the SV of the current step as fast as possible.

This safety-start feature is useful when the user is very strict about the temperature and the time of a critical step. It is also useful when the user does not want to control the ramp speed and want the system to reach the set temperature as soon as possible. Please see section 8.2 for examples.

During this period when program was put to "hold", the relay(s) assigned to deviation alarms (Hy-1 and Hy-2) will not be activated; but relay(s) can still be activated by process alarms (ALM1 and/or ALM2).

**Safety range (deviation range):** A temperature range defined by deviation alarms and hysteresis band. The lower boundary is (SV - Hy-2 + Hy), and the upper boundary is (SV + Hy-1 - Hy). The safety range / deviation range has effects on program execution and deviation alarms.

At the beginning of a step or at powering up the controller, the program will be put on "hold" if PV is out of the safety range; if then PV falls within the safety range, safety-start will be granted. Once a step starts running, it won't be put to "hold" even if PV falls out of the deviation range again.

When a program step is running, deviation alarms will be triggered if PV is greater than (SV + Hy-1 + Hy) or less than (SV - Hy-1 - Hy), and the corresponding relay(s) will pull in. When PV falls within the safety range, i.e. PV is less than (SV + Hy-1 - Hy) and greater than (SV - Hy-2 + Hy), the alarm will be deactivated and the corresponding relay will drop out.

**6.2. Program Setup**

Press the A/M key to bring the controller into the program setup mode. The controller will show the current step number in the top window "C XX" and show the set temperature in the lower window. The default value is -1 (displayed as "- 1"). Use the A/M key to choose which digit to edit (indicated by a flashing decimal point), and use the UP or DOWN arrow key to adjust the set temperature (-1999 to +9999). Then press SET key to save the change and go to the step set time. The top window will show "t XX", and the lower window will show the default time -1 (displayed as "- 1"). Use the UP, DOWN, or A/M key to change the value, then press SET key to confirm and go to the next step. At each program step, the set temperature and the set time is displayed in turn. Repeat this operation until all steps in your program have been entered. You don't need to use/edit all 30 steps.

**Note 1:** The above operation is inhibited if the program setup function is locked (refer to 5.3.16 for the details of the Lock parameter).

**Note 2:** In parameter setting mode or programming mode, to go back to the previous parameter, hold the A/M key and press the DOWN arrow key (V). To exit the program editing mode, (1) hold A/M key and press SET key, or (2) leave the key pad untouched for 10 seconds.

### 6.2.2. Step Set Time

The set time between step  $n$  and step  $(n + 1)$  is specified by  $t[n]$ . The step time can be set to any value from 1 to 9999, the unit is in minutes. The step time  $t[n]$  can also be assigned with a negative value to perform a special command such as jumping to a specified step or activate an alarm relay (see section 6.2.10 for details).

### 6.2.3. Program Ramp

Ramp means change the temperature from one point to another during a specified time. To program a ramp, you need to set the start temperature  $C[n]$ , the end temperature  $C[n+1]$ , and the time duration  $t[n]$ . The ramping speed is

$$V = (C[n+1] - C[n]) / t[n].$$

During this ramping step, the displayed set temperature will linearly change from  $C[n]$  to  $C[n+1]$  in proportion to the time that has been past in this step. For example, at step 3, you want the controller to ramp up from 200 to 300 degrees in 50 minutes, then you'll need to set  $C03 = 200^\circ\text{F}$ ,  $t03 = 50\text{min}$ , and  $C04 = 300^\circ\text{F}$ . The ramp up speed is  $2^\circ\text{F}/\text{min}$ . If this step has been run for 10min, the set temperature is  $220^\circ\text{F}$  at this moment.

The ramping speed should be less than or equal to the maximum ramping speed that the current system can achieve at full power. In other words, the ramping time should be longer than the minimum time needed for the system to jump from  $C[n]$  to  $C[n+1]$ . Otherwise, the process temperature will gradually fall behind the schedule. For example, the maximum heating speed of an electric water kettle is  $20^\circ\text{F}/\text{min}$ , but the ramping step is set as  $C01 = 90^\circ\text{F}$ ,  $t01 = 1\text{min}$ , and  $C02 = 212^\circ\text{F}$ . The controller will not be able to achieve this goal. In the end of this step, the water temperature is probably at  $110^\circ\text{F}$ .

At the beginning of every step, the controller will check whether the process temperature is within the safety range (see section 6.1 for the definition). If temperature falls within the safety range, the program will continue, and the timer for this step will be started. If the temperature is out of the safety range, the program will pause, controller will keep on trying to reach the set temperature, and the timer won't be started until this condition is satisfied; in the meantime, the deviation alarm is triggered. This feature is called "safety-start". This feature only applies to the beginning of each step. If a step has already been running, even if the temperature falls out of the safety range, the program will still continue. But the deviation alarm will be triggered.

If the maximum speed of a system is unknown or varies with environmental conditions, users can use the "safety-start" feature to ensure that the temperature and the time during a ramp/soak step are kept within a reasonable range required by the process. This is done by setting the deviation alarms close to the SV. In another ward, keep the deviation range small.

For example, a user want to preheat a small oven from room temperature to  $400^\circ\text{F}$  and maintain it for 10min before preceding to the next step; the acceptable deviation is  $\pm 3^\circ\text{F}$ . So the program can be set as  $C01 = 400$ ,  $t01 = 10$ ,  $C02 = 400$ ,  $t02 = xx$ . And set  $Hy-1 = 3$ ,  $Hy-2 = 3$ , and keep the  $Hy = 0.3$  (default value). So the lower boundary of the deviation range is  $398.7^\circ\text{F}$  and the upper boundary of the deviation range is  $402.7^\circ\text{F}$ . After the program is started, the controller will start bringing up the temperature. Once the process temperature reaches  $398.7^\circ\text{F}$ , the controller will start the step1 and counting down for 5 minutes. Please see section 8 for more examples.

### 6.2.4. Program Soak

The soak segment maintains the temperature for a specified time. It can be considered as a special case of ramping with a zero degree slope. To program

a soak, you need to set the start and the end temperature to be the same, i.e.,  $C[n] = C[n+1]$ , and the soaking time is specified by  $t[n]$ . For example, at step 3, the user wants the parts to be soak in the oven at  $200^\circ\text{F}$  for 60 minutes, then the program for this step should be  $C03 = 200$ ,  $t03 = 60$ , and  $C04 = 200$ .

**Note:** The step time is not how long the controller will stay at the set temperature for the current step. It is how long the controller will take from the current step set temperature to the next step set temperature. These two concepts are very different.

### 6.2.5. Program Run

When the program is running, the A-M indicator should be on. To start running a program or resume a program that is on "hold", press the DOWN arrow (V) key for 2 seconds until the lower window shows "r u n".

### 6.2.6. Program Hold

"Hold" means the program is being temporarily stopped. When the program is on hold, the A-M indicator will blink, the lower display window will flash "H o l d" and set temperature. Under any of the following situations, the program will be put on "hold":

- 1) Step time  $t[n] = 0$ .
- 2) A jump step transits to another jump step.
- 3) Program is manually put into "hold" by pressing DOWN arrow (V) key for 2 seconds.
- 4) A step is programmed to jump to itself, i.e.,  $t[n] = -n$ .

For the situation (1) ~ (3), user can hold the DOWN arrow (V) key for 2 seconds to resume the program. For the 4th situation, holding the DOWN arrow key won't resume the program; but user can manually jump to another step.

### 6.2.7. Program Stop

When a program is stopped, the step number is reset to 1, the event output is cleared, and the control output is turned off. The lower display window will show "S t o p", and the A-M indicator is off.

The program can be stopped by the following methods:

- 1) Step time  $t[n] = -121$ .
- 2) Manually stop the program by press the UP arrow ( $\wedge$ ) key for 2 seconds until the lower display window shows "stop". When the program is stopped, the lower window will flash "S t o p" and the current set temperature.

### 6.2.8. Program Jump

The program can go to a specified step. To jump the program from the current step  $n$  to step  $x$ , set  $t[n] = -x$ , where  $x$  is the step number and it can be any integer between 1 and 30. The minus sign is a special indicator telling the controller this is a program command, not a step time.

**Note:** If a step is programmed to jump to itself (i.e.,  $t[n] = -n$ ), it would result in a "Hold" status that will never be released. For example if  $t06 = -6$ , the program will be held at step 6.

### 6.2.9. Commands for Special Event

Special events such as pulling in or dropping off alarm relays, jumping to another step, holding the program, or stopping the program can be programmed by assigning negative values to the set time  $t[n]$ . When  $t[n]$  is between 1 and 9999 (min), it is used to set the ramp and soak time; when  $t[n]$  is set to zero, controller will be put in hold mode on step  $n$  until manually released by the operator; when  $t[n]$  is set to a negative number, it is used for executing other special commands.

To program a special event, assign set time  $t[n] = x$ , where  $x$  is a negative integer between -240 and -1. The value is calculated by the equation:

$$t[n] = -(A \times 30 + B),$$



where A is the event (ranging from 0 to 7) and B is the step number (ranging from 1 to 30) to which the program will jump.

Each value of A represents an event:

- A = 0, no special effect, for jumping steps only;
- A = 1, switch on AL2;
- A = 2, switch on AL1;
- A = 3, switch on AL1 and AL2;
- A = 4, stop the instrument (B must be set to 1 when A = 4);
- A = 5, switch off AL2;
- A = 6, switch off AL1;
- A = 7, switch off AL1 and AL2.

**Examples:**

- Jump from step 4 to step 5 and switch on AL2:  
 $t_{04} = -(1 \times 30 + 5) = -35$
- Jump from step 6 to step 1 and switch off AL2:  
 $t_{06} = -(5 \times 30 + 1) = -151$
- Stop the program at step 8:  
 $t_{08} = -(4 \times 30 + 1) = -121$

**Note:** If a step is programmed to jump to itself (e.g.,  $t_{06} = -6$ ), it would result in a "hold" status that will never be released. The operator need to manually stop the program or jump to another step.

**6.2.10. The Last Step**

To completely stop a program, set  $t[n] = -121$ . This command should be placed at the end of the program. It will turn off all outputs from the controller and stand by at step 1.

At step 30, if the controller wasn't programmed to stop or jump, it will continue to execute step 1 after the current step is finished.

**6.2.11. Check the Step Number and Jump the Program**

To check at which step the program is running, press the SET key once. The upper window will show "STEP" and the lower window will show the step number. Press SET key again, the controller will show the set time for the current step in the upper window, and show the running time in the current step in the lower window. Press the SET key twice to return to the normal operating mode. If no key is pressed in 10 seconds, it will automatically return to the normal operating mode. While checking the step number or time, the step number or time will be updated as the program executes.

To manually jump to another step in the program, press the SET key once to show the step number in the lower window, then press the UP arrow (Λ) or DOWN arrow (V) key to change the step number, and press the SET key to confirm the change. The program will be jumped to the specified step immediately. The controller will show the step set time for the new step in the upper window. And the running time in the lower window will be cleared to 0. If the step number is not changed, press SET key won't affect the execution of the current step, but the display window will show step set time and running time.

This is a convenient feature when you want to skip the current step and manually jump to another step. For example, if you want to jump to step 5 while the program is running, press SET key to check the step number, then change the step number to 5, and then press SET key to confirm.

**6.2.12. Multiple Program Segments**

The controller can save up to 30 program steps, which gives user the flexibility to store multiple short programs and select which segment of program to run. User can manually select a starting step to run the program, or program a jump step in step 1. A command to stop the program ( $t[n] = -121$ ) can be programmed at the end of the each short program. Please refer to the example program below.

Table 6. Program example.

Step #	C[n]	t[n]	Note
1	80	-5	Jump to step 5.
2	200	30	Program 1 (step 2 to 3).
3	200	-121	Turn off all output and stand by at step 1.
4	-1	-1	
5	800	30	Program 2 (step 5 to 6).
6	800	-121	Turn off all output and stand by at step 1.
7	80	20	Program 3 (step 7 to 13).
8	400	60	
9	1500	20	
10	1500	90	
11	800	60	
12	100	60	
13	100	-121	Turn off all output and stand by at step 1.
14	-1	-1	
15	-1	-1	
...	...	...	...
30	-1	-1	

In the example above, there are three short programs stored. Program1, step 2 to 3; program2, step 5 to 6; program3, step 7 to 13. The step time of step 1 can be set as follows to choose the desired program:

- $t_{01} = -2$ , jump to step 2 and run Program 1.
- $t_{01} = -5$ , jump to step 5 and run Program 2.
- $t_{01} = -7$ , jump to step 7 and run Program 3.

You can also choose the program by manually setting the step number before the program starts. For example, if Program 2 is needed in the current process, press SET key to show step number, then change the step number to 5, and then press SET key again to confirm. (See detailed instruction in section 6.2.11.)

**7. Wiring Examples**

**7.1. Controlling the load directly with the internal relay**

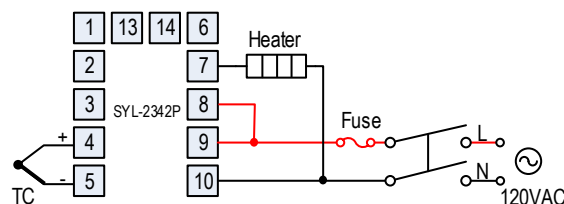


Figure 6. Controlling a heater by SYL-2342P.

This is a simple set up where the heater is directly controlled with the internal relay of the controller. The heater must consume less current than the internal relay's maximum rating (7A at 240VAC and 10A at 120VAC). The temperature sensor in this example is a thermocouple.

**7.2. Controlling the load via external contactor**

Using the external contactor allows users to control higher power loads than the internal relay can handle. It is also easy to service. If the contacts of the relay wear out, it is more economical to replace them than to repair the controller. In this example, we assume the coil voltage of the contactor is the same as the voltage of the controller power supply. The voltage of power supply for alarm is 120V AC. **Note: You don't have to wire or set the alarm to control the temperature. It is just to show how the alarm can be wired.**

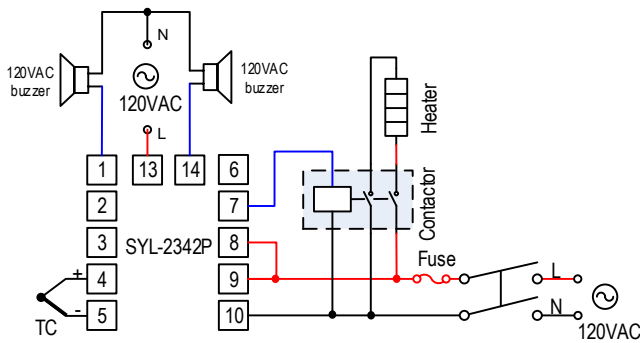


Figure 7. SYL-2342P with thermocouple input and external relay output. This is a typical wiring for oven and kiln.

**7.3. Controlling a 24V valve**

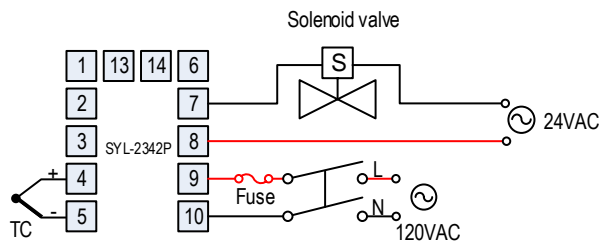


Figure 8. Controller a solenoid valve with SYL-2342P.

This is a typical wiring set up for a 24V gas valve, hot water valve, or a contactor with 24V coil voltage. A separate 24V power is needed to drive the valve. Temperature is measured with a thermocouple input on pin 4 and 5.

**7.4. Cooling and heating with the same controller**

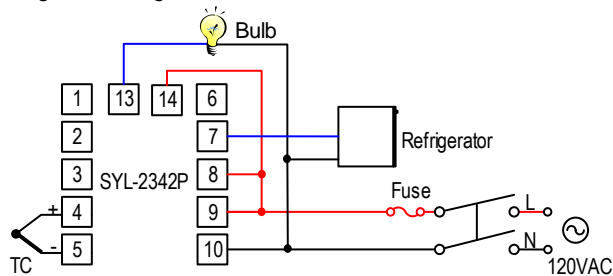


Figure 9. A refrigerator connected to SYL-2342P directly.

This is an example for beer fermentation. The refrigerator is driven by the internal relay of the controller directly. **Please note that the internal relay output (terminals 7 and 8) is not powered by itself.** A power supply **must** be used to drive the external relay. The refrigerator must consume less current than the internal relay's maximum rating (7A at 240V AC and 10A at 120V AC). The bulb (less than 100 - 300 W) is for heating. The example is setup to have the heater to turn on when temperature drops to below 60 °F and turns off at 64 °F. The refrigerator will turn on when temperature is above 69 °F and turn off at 65 °F.

**Set up for the controller:**

- 1) Hy = 2.0. Set both hysteresis band for heater and cooler to 2 degree.
- 2) COOL = 9. Set the controller to cooling mode, no alarm suppression, Fahrenheit temperature unit display.
- 3) AT = 0. Set the controller main output to on/off control mode for refrigerator compressor control
- 4) ALM2 = 62. Set the low limit alarm to 62 °F. Heater will be on at 60 °F (ALM2-Hy) and off at 64 °F (ALM2+Hy)
- 5) SV = 67. Refrigerator will be on at 69 °F (SV + Hy) and off at 65 °F (SV - Hy).

**7.5. Maintaining a temperature difference using two thermocouples**

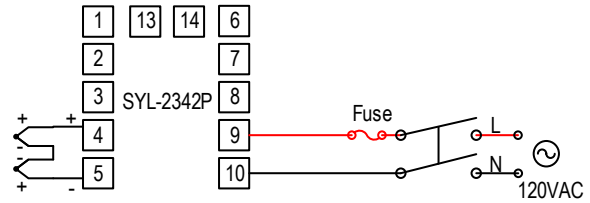


Figure 10. SYL-2342P with two thermocouple inputs to measure temperature difference.

Connect two thermocouples in series with opposite polarity (negative connected to negative). Leave the two positive connected respectively to the input terminals on the controller. The one for lower temperature is connected to negative input of the TC input. The one for higher temperature is connected to the positive input.

**Set up the controller (assuming type K thermocouple is used):**

- 1) Sn = 35. Set the input type to -20mV ~ +20mV. It eliminates the interference of internal cold junction compensation circuit.
- 2) P-SL = -485 and P-SH = 485. This converts the millivolt signal to Celsius degrees. (P-SL = -905 and P-SH = 905 for Fahrenheit degrees). To maintain a 20 °C difference, set SV = 20.

**Note:** For a type K thermocouple, the voltage vs. temperature plot is not perfectly linear over its full working range. When calculating the P-SL and P-SH for this application, we found the corresponding temperature of 20mV signal assuming the reference junction is 0 °C. Please contact us if you have any question.

**7.6. Controlling the load via external contactor in a 240V system.**

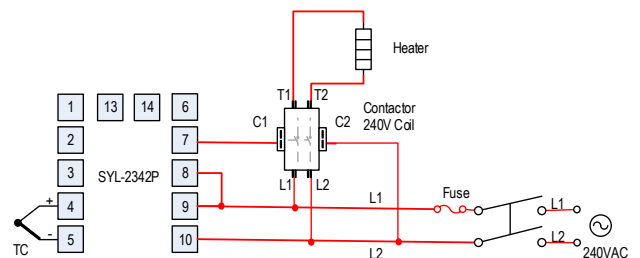


Figure 11. SYL-2342P with an external contactor to control the heating element in a 240V system.

This example is very similar to example 7.2, but both the controller and the contactor's coil are drove by 240VAC power.

**8. Programming Examples**

Programs in the SYL-23X2P series controller have a uniform format of temperature-time-temperature. The temperature set point of the current step will linearly ramp to the set point of the next step over the time interval between the two steps. Usually, it is not recommended to program a ramp step that is faster than what is the system is capable of achieving because it will result in actual temperature fall behind the schedule **UNLESS** you are using the Safety-Start feature (see section 6.1 for its definition and see Figure 12 for an example). The time units are in minutes. Negative values of the time interval represent program commands.

**8.1. Example 1: holding the oven temperature at 800°C for 2 hours.**

Assuming that the heater is able to heat the oven from 25°C to 800°C within 30 minutes, we can write a program like this:

**Step 1: C01 = 25, t01 = 30.** Start linear temperature heating up from 25°C to 800°C, over a time period of 30 minutes (25.8 °C /minute).

**Step 2: C02 = 800, t02 = 120.** Maintain 800 °C for 120 minutes.

**Step 3: C03 = 800, t03 = -121.** Stop the program and let the oven cool down. The equation used to get the command number is:  $-(30 * \text{Command\#} + \text{Next Step}) = -(30 * 4 + 1) = -121$ .

The temperature profile is shown below in Figure 12.

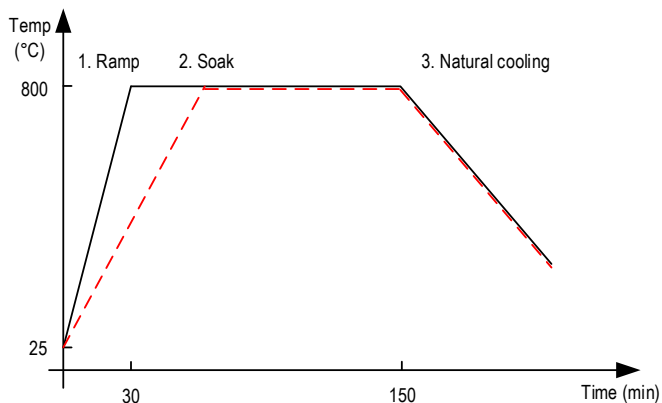


Figure 12. Temperature profile of an oven that is programmed to hold temperature at 800 °C for 2 hours. The solid line is the temperature when the oven is capable of ramping up from 25 °C to 800 °C in 30 min; the dashed line is the temperature profile when the oven needs longer time to ramp up to 800 °C.

**Note 1:** The value of C[n] is the beginning temperature of step n. For instance, C01 is always the temperature at the beginning of step 1, and C02 is the beginning temperature of step 2. Usually C01 should be the ambient temperature, and t01 is the time from step 1 to step 2.

**Note 2:** If the heater is not capable of heat the oven from 25°C to 800°C in 30min, the actual oven temperature can fall behind the schedule. The timer for step 2 (the soak section) will start even when temperature is still below 800°C; and so the actual soaking time will be less than 2 hours. Please see the dashed line in Figure 12.

**8.2. Example 2: program alarm relay actions**

The following example includes 6 steps: linear temperature heating, maintaining a constant temperature, linear temperature cooling, jump cycling, ready and event output. In the following example, it is assumed that the deviation high alarm  $Hy-1 = Hy-2 = 5^{\circ}\text{C}$  and  $Hy = 0$ .

**Step 1: C01 = 100, t01 = 30.** Start linear temperature heating up from 100°C to 400°C in a time period of 30 minutes (ramp speed is 10°C /minute).

**Step 2: C02 = 400, t02 = 60.** Maintain 400°C for 60 minutes.

**Step 3: C03 = 400, t03 = 120.** Reduce the temperature from 400°C to 160°C in 120min. So the ramp speed is:  $(C04-C03) / t03 = -2^{\circ}\text{C}/\text{minute}$ .

**Step 4: C04 = 160, t04 = -65.** Activate relay AL1 and go to step 5. This command is calculated by plugging  $A = 2$  and  $B = 5$  into the expression  $t[n] = -(A * 30) + B = -(2 * 30 + 5) = -65$ .

**Step 5: C05 = 160, t05 = 0.** Hold (pause) the program at step 5. Operator/user needs to press the DOWN arrow key to resume running the program.

**Step 6: C06 = 100, t06 = -181.** Switch off relay AL1 and jump back to step 1. (Please note that the AL1 relay can still be activated by temperature alarms.) This command for switching AL1 off is calculated by plugging  $A = 6$  and  $B = 1$  into the expression  $t[n] = -(A * 30) + B = -(6 * 30 + 1) = -181$ .

**Note 1:** When the program jumps back to step 1 ( $C01 = 100, t01 = 30$ ), the oven temperature is still at 160 °C, the program will pause until the temperature drops within the deviation alarm range of the new set point. As the deviation high alarm is set to 5°C, the program will resume (from the beginning) as soon as the temperature drops below  $SV + Hy-1 - Hy = 105^{\circ}\text{C}$ .

**Note 2:** The current step 5 can be omitted, i.e., we can change the step 5 to  $C05 = 160, t05 = -181$ . When two jumps steps are programmed next to each other, the program will pause.

The temperature control block is shown below.

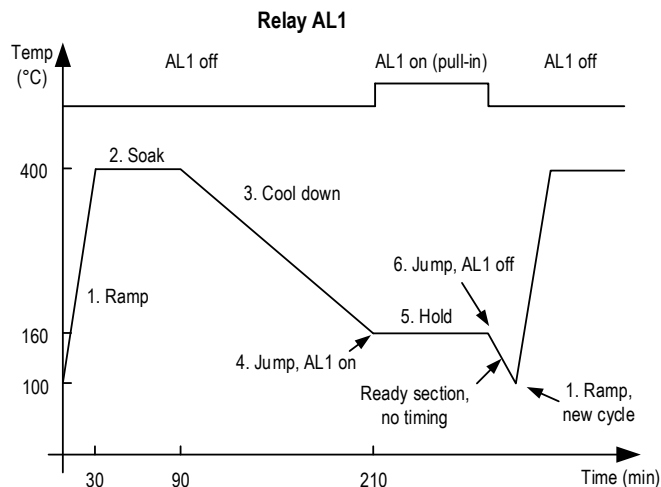


Figure 13. Temperature profile and relay AL1 action in the example program.

**8.3. Example 3: powder coating oven**

A powder coating oven needs to heat up to 375°F at its maximum speed. Then hold the temperature for 20 minutes. After that, pull in the AL2 relay for 2 minutes. If the temperature is higher than 450°F, trigger the AL1 relay, and if the temperature is lower than 446°F, deactivate the AL1 relay. This application needs a controller SYL-2342P, a 40A/240V contactor, a thermocouple and a two buzzers with a 12V DC power source.

**a. Wiring**

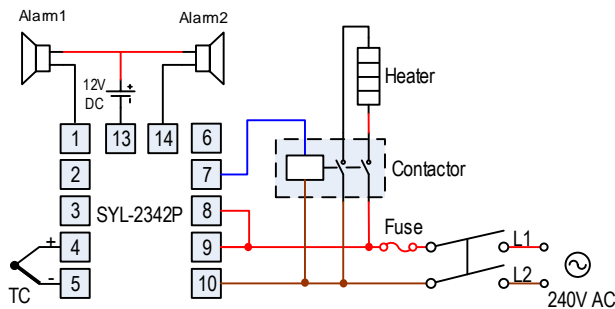


Figure 14. Wiring diagram for the powder coating oven in example 3.

**b. Parameter setting and program**

There are the parameters that need to be changed from the default value:  
ALM1 = 448, Hy = 2, Hy-1 = 3, Hy-2 = 3. (Note: **Hy value should be smaller than both Hy-1 and Hy-2.**) By default AL-P = 17, so the high limit alarm is already assigned to relay AL1 as required by this application.

The program:

**C01 = 375, t01 = 20.** Soak at 375°F for 20 minutes. The system will heat up from the ambient temperature to 374 °F (SV - Hy-2 + Hy) before the timer starts.

**C02 = 375, t02 = - 33.** Turn on relay AL2 and jump to the next step.

**C03 = 375, t03 = 2.** Hold temperature at 375°F, relay AL2 stays on.

**C04 = 375, t04 = - 121.** Stop the program. Heater off, relay drop off. Oven cools down at its natural cooling speed.

**Auber Instruments Inc.**

5755 North Point Parkway, Suite 99,

Alpharetta, GA 30022

[www.auberins.com](http://www.auberins.com)

Email: [info@auberins.com](mailto:info@auberins.com)

Copyright © 2022 Auber Instruments Inc. All rights reserved.  
No part of this datasheet shall be copied, reproduced, or transmitted in any way without the prior, written consent of Auber Instruments. Auber Instruments retains the exclusive rights to all information included in this document.