Classification	ISaGRAF English FAQ-142						
Author	Chun Tsai	Version	1.1	Date	Jun.2011	Page	1 / 1

How to protect your ISaGRAF program from used by the unauthorized people?

When you have developed your ISaGRAF application ready to deliver to your customers, please think about whether your application within the ISaGRAF PAC will be copied to another same model of PAC. Be careful, most of the third parties steal your hard work effort in this way. Here provide a simple and easy way to protect your applications.

NOTE:
If you give the ISaGRAF source codes to your customers, sorry, the following method cannot effectively protect your projects from stolen, because having the source code, the thief can modify the source code to applied to another PAC.

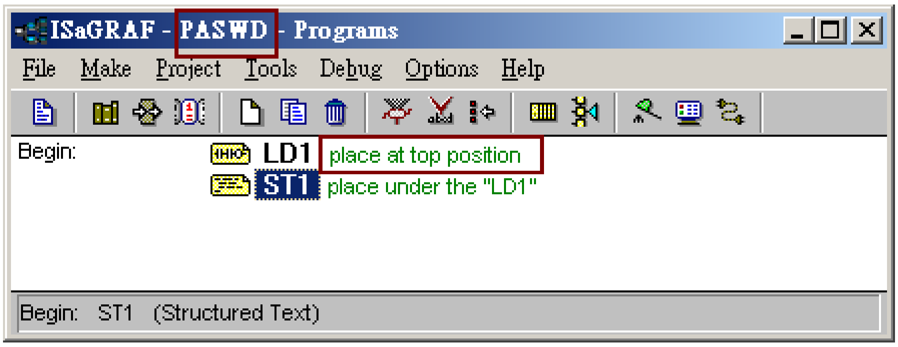
Each ICP DAS ISaGRAF PAC has a unique 8-byte (or 64-bit) serial number different from other PACs. You can use this serial number through a special algorithm to generate a password, and save this password in the EEPROM of PAC, then make a verify test. If the program cannot pass the verification, the application will not work.

The following example uses two ISaGRAF projects, one named “paswd” which is used to generate the PAC password and store it to the EEPROM, another named “my_ap” which is the developed application ready to give to the customer. Please download and run the program “paswd” to the PAC to generate an exclusive password for that PAC, and then download the application “my_ap” to the same PAC before delivering it to the customer. After this process, even thought the ISaGRAF application is copied to a same model PAC, the application will not work because it cannot pass the verification.

The Application Program “paswd”:

First, use “Get_SN” to read the serial number. Next, generate a password by a user defined algorithm. Then, store the password to the EEPROM at a user decided address.

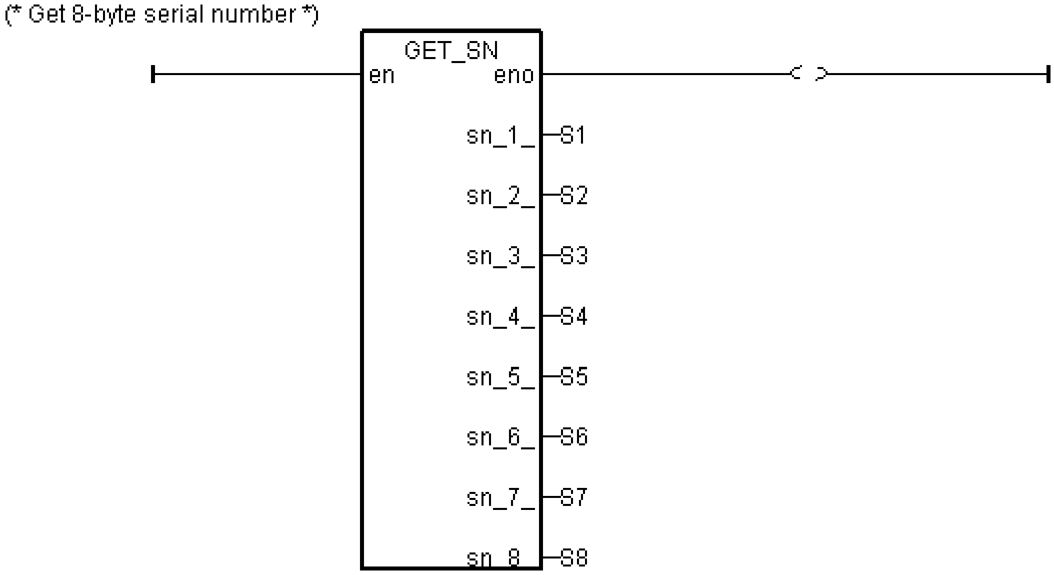
Project :


ISaGRAF - PASWD - Programs
File Make Project Tools Debug Options Help
Begin:
LD1
place at top position
ST1
place under the "LD1"
Begin: ST1 (Structured Text)

Variables :

INIT : Boolean , **set initial value at TRUE**
TMP : Boolean, temporary usage
S1 ~ S8 : integer, to get the 8-byte serial number
password : integer, the password to be generated.

Program – LD1 :



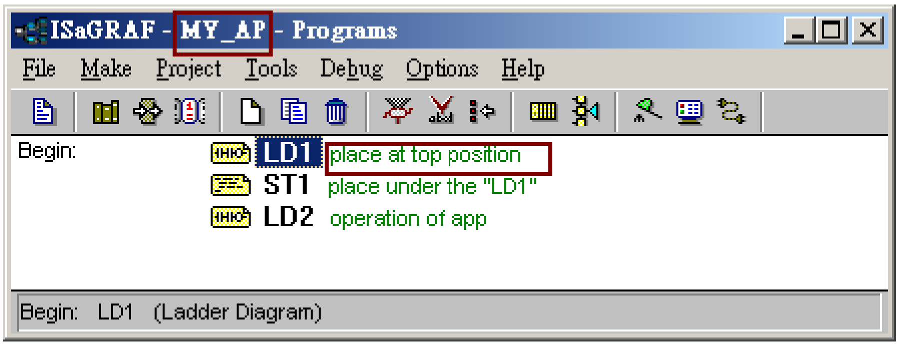
Program – ST1 :

```
(* Please must declare "INIT" as Boolean variable with an initial value - TRUE *)
if   INIT   then
   (* run only once in the 1st PLC scan cycle *)
   INIT := False ;
   (* Please must use your algorithm to generate password.   This demo only generate one
      password (32-bit integer). You may choose to generate two or more passwords.    *)
   password := (S1 * S2) + (S3 * 12345) + S4 + (S5 * S6 * S7) + S8 ;
   (* store the password to the EEPROM *)
   TMP := EEP_EN( ) ;    (* remove write-protection of the EEPROM *)
   (* please must modify it to store your password in a different address *)
   TMP := EEP_N_W( 261 , password ) ; (* store the password to EEPROM *)
   TMP := EEP_PR( ) ;    (* protect the EEPROM *)
end_if ;
```

The Application Program “my_ap”:

First, use “Get_SN” to read the serial number and also read the password from EEPROM. Next, verify if the password is correct, if not, let the application cannot work.

Project :


ISaGRAF - MY_AP - Programs
File Make Project Tools Debug Options Help

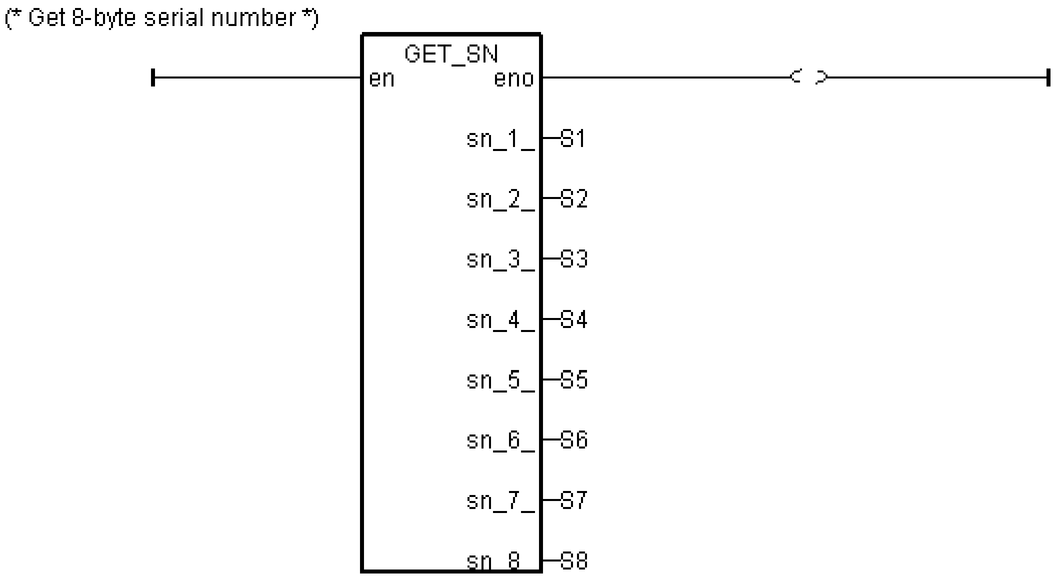
Begin:
LD1 place at top position
ST1 place under the "LD1"
LD2 operation of app

Begin: LD1 (Ladder Diagram)

Variables :

INIT : Boolean , **set initial value at TRUE**
TMP : Boolean, temporary usage
password_ok : Boolean. False: invalid password; True: correct password.
S1 ~ S8 : integer, to get the 8-byte serial number
password : integer, the password to be generated.

Program – LD1 :

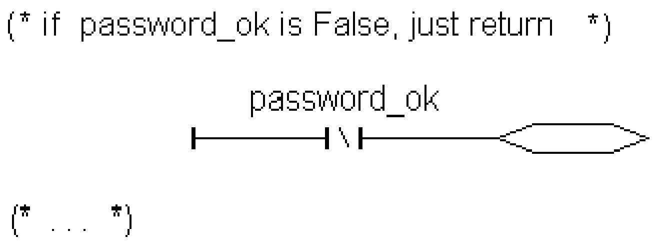


Program – ST1 :

```
(* Please must declare "INIT" as Boolean variable with an initial value - TRUE *)
if   INIT   then
   (* run only once in the 1st PLC scan cycle *)
   INIT := False ;
   (* read password *)
   password := EEP_N_R( 261) ;
   (* Please must use your own algorithm to generate your password. *)
   password_ok := False ;
   if   password = ( (S1 * S2) + (S3 * 12345) + S4 + (S5 * S6 * S7) + S8 )   then
        password_ok := TRUE ;
   end_if ;
end_if ;

(* Then user can use this Boolean variable - "password_ok" in other programs
     in the same project to protect your application *)

   if     password_ok = False    then
           return ;
  end_if ;

(* ... *)
```

If “password_ok” is False, all the programs after will be return. (This example uses a LD2 only.) If there are other programs in your application, please add the “password_ok” code for each as below.

Program – LD2 :



```
(* if  password_ok is False, just return   *)

(*  . . .  *)
```