**Proyecto de Informática Industrial 1**

**Título: Invernadero**

**Autores: Sergio Bou Grau**

**Carlos Martín Juárez**

**Jose Antonio Mengual Sanchez**

Fecha de entrega: 19/12/2019

**Índice**

1. Introducción.
   1. Descripción del proyecto.
   2. Objetivos
2. Fichero de cabeceras del proceso (proceso.h).
3. Fichero de cabeceras del data (datosComunes.h).
4. Fichero de cabeceras del usuario (usuario.h).
5. Capturas de pantallas y explicación de la aplicación desarrollada.
6. **Introducción**
   1. **Descripción del proyecto.**

Se ha realizado un proyecto para el control automatizado de un invernadero, este control tiene una interface con el usuario sencilla y ágil. Para la medida de las características físicas de importancia se ha empleado unos sensores de tipo digital (2 estados, alto y bajo) y analógicos (múltiples estados). El control tiene una serie de indicaciones, que gracias a ellas vemos el estado en el que se encuentra el sistema. Además es capaz de detectar anomalías que son perjudiciales para el cultivo, actuando sobre ellas y mostrándolas en el panel de control.

* 1. **Objetivos.**

El objetivo de este proyecto es la creación de un programa para el control de cultivo de un invernadero mediante una programación bien estructurada (por módulos) y fácil de leer; realizar la lectura y escritura de los datos digitales y analógicos en la entrada/salida de la tarjeta de adquisición de datos, además del manejo del programa C++ Builder 6 y de la tarjeta de adquisición de datos NI USB – 6008 (Digital).

1. **Fichero de cabeceras del proceso (proceso.h)**

//PROCESO H

1. *//--------------------------------------------------------------*
2. #ifndef ProcesoH
3. **enum** {PIN\_OFF, PIN\_ON};
4. **void** process\_error();//Muestra los mensajes de error
5. **void** process\_init(**char**\*);//Crea las tareas y los canales
6. **void** process\_read\_port1(**void**);//Lee el puerto 1
7. **void** process\_read\_ai0(**void**); //Lee el puerto Ai0
8. **void** process\_read\_ai1(**void**); //Lee el puerto Ai1
9. **void** process\_write\_port0(**void**); //Escribe en el puerto 0
10. **void** process\_write\_ao0(**void**); //Escribe en el puerto Ao0
11. **void** process\_write\_ao1(**void**); //Escribe en el puerto Ao1
12. #define ProcesoH
13. *//---------------------------------------------------------*
14. #endif

**3. Fichero de cabeceras del data (datosComunes.h)**

1. // DATA H
2. *//---------------------------------------------------------*
3. #ifndef DatosComunesH
4. #define DatosComunesH
5. **void** Store\_Port1(**int**);//Almacena el valor del puerto 1.
6. **int** estado\_Port1(**void**);//Devuelve el valor del puerto 1.
7. **void** Store\_AI0(**float**);//Almacena el valor del puerto AI0.
8. **double** estado\_AI0(**void**);//Devuelve el valor del puerto AI0
9. **void** Store\_Port0(**int**, **int**);//Almacena el valor delpuerto0.
10. **int** estado\_Wport1(**void**); //Devuelve el valor del puerto 1
11. **int** estado\_Pin(**void**); //Devuelve si el pin es H o L.
12. **double** EstadoHumedad(**void**); //Devuelve el valor del nivel de humedad
13. **void** Store\_Humedad(**double**); //Almacena el nivel de humedad
14. **void** Store\_AI1(**float**); //Almacena el valor de puerto AI1
15. **double** estado\_AI1(**void**); //Devuelve el valor de puerto AI1
16. **void** Store\_Fan(**double**); //Almacena el nivel del ventilador
17. **double** EstadoVentilador(**void**); //Devuelve el nivel del ventilador
18. *//---------------------------------------------------------*
19. #endif

**4. Fichero de cabeceras del usuario (usuario.h)**

1. USUARIO H
2. *//--------------------------------------------------------------*
3. #ifndef UsuarioH
4. #define UsuarioH
5. *//--------------------------------------------------------------*
6. #include <Classes.hpp>
7. #include <Controls.hpp>
8. #include <StdCtrls.hpp>
9. #include <Forms.hpp>
10. #include <ExtCtrls.hpp>
11. #include <Graphics.hpp>
12. *//---------------------------------------------------------------------------*
13. **double** redondeo(**double**);
14. **class** TVPrincipal : **public** TForm
15. {
16. **\_\_published**: *// IDE-managed Components*
17. TTimer \*TimerEstadoPuertos;
18. TPanel \*PIniciarT;
19. TPanel \*PTimer;
20. TLabel \*Label3;
21. TEdit \*Edit1;
22. TTimer \*TimerLedHumedad;
23. TGroupBox \*GroupBoxEntradas;
24. TLabel \*Label8;
25. TImage \*ImageNoche;
26. TImage \*ImageDia;
27. TImage \*ImagePCerrada;
28. TImage \*ImagePAbierta;
29. TCheckBox \*CheckBoxDia;
30. TCheckBox \*CheckBoxPuertaAbierta;
31. TEdit \*Edit2;
32. TCheckBox \*CheckBoxNoche;
33. TCheckBox \*CheckBoxPuertaCerrada;
34. TTimer \*TimerPAbierta;
35. TLabel \*Label16;
36. TEdit \*Edit3;
37. TShape \*Shape7;
38. TShape \*Shape8;
39. TShape \*Shape9;
40. TShape \*Shape10;
41. TImage \*Image1;
42. TImage \*Image2;
43. TGroupBox \*GroupBox1;
44. TLabel \*Label9;
45. TShape \*Shape4;
46. TLabel \*Label10;
47. TLabel \*Label11;
48. TShape \*Shape5;
49. TLabel \*Label12;
50. TEdit \*ContadorPA;
51. TEdit \*ContadorHU;
52. TGroupBox \*GroupBoxSalidas;
53. TShape \*Shape1;
54. TShape \*Shape2;
55. TShape \*Shape3;
56. TLabel \*Label5;
57. TLabel \*Label6;
58. TLabel \*Label7;
59. TLabel \*Label13;
60. TLabel \*Label14;
61. TLabel \*Label15;
62. TShape \*Shape6;
63. TLabel \*Label18;
64. TLabel \*Label19;
65. TCheckBox \*CheckBoxLampara;
66. TCheckBox \*CheckBoxClima;
67. TCheckBox \*CheckBoxValve;
68. TCheckBox \*CheckBoxFan;
69. TScrollBar \*ScrollBarElectroValvula;
70. TEdit \*Edit4;
71. TEdit \*Edit5;
72. TScrollBar \*ScrollBarVentilacion;
73. TShape \*Shape11;
74. TShape \*Shape12;
75. TShape \*Shape13;
76. TShape \*Shape14;
77. **void \_\_fastcall** BotonIniciarTarjeta(TObject \*Sender); //Activa la tarjeta de adquisición de datos
78. **void \_\_fastcall** BotonIniciarAutomata(TObject \*Sender);

//Activa el temporizador de proceso

1. **void \_\_fastcall** TimerPuertos(TObject \*Sender);

//Activa el temporizador para leer los puertos.

1. **void \_\_fastcall** Timer\_Led\_Humedad(TObject \*Sender);

//Parpadeo del led de humedad.

1. **void \_\_fastcall** CheckBoxDiaClick(TObject \*Sender);

//Marca es de dia.

1. **void \_\_fastcall** CheckBoxNocheClick(TObject \*Sender);

//Marca es de noche.

1. **void \_\_fastcall** CheckBoxPuertaCerradaClick(TObject \*Sender);//Marca puerta cerrada
2. **void \_\_fastcall** CheckBoxPuertaAbiertaClick(TObject \*Sender); //Marca puerta abieta.
3. **void \_\_fastcall** TimerPuertaAbierta(TObject \*Sender);

//Determina la alarma de puerta abierta.

1. **void \_\_fastcall** CheckBoxLamparaClick(TObject \*Sender);

//Activa la lámpara.

1. **void \_\_fastcall** CheckBoxClimaClick(TObject \*Sender);

//Activa el climatizador.

1. **void \_\_fastcall** CheckBoxValveClick(TObject \*Sender);

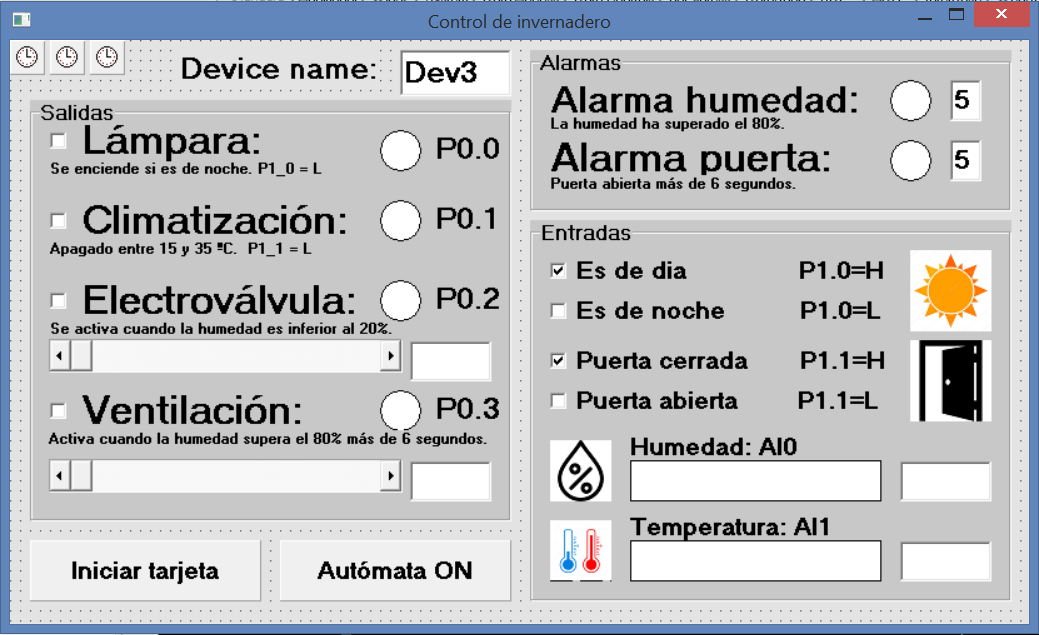
//Activa la electroválvula.

1. **void \_\_fastcall** CheckBoxFanClick(TObject \*Sender);

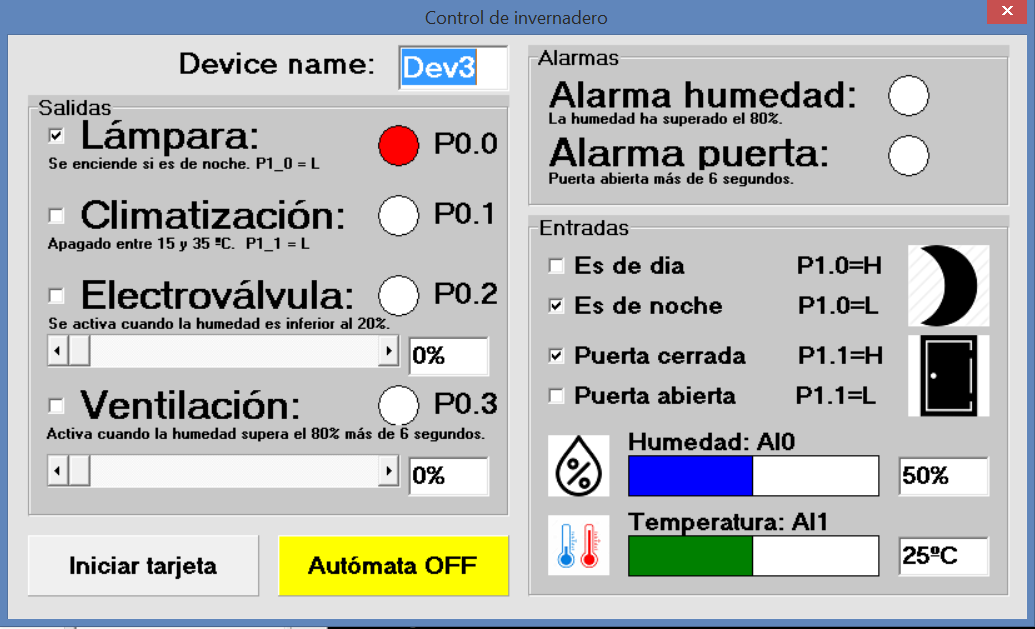
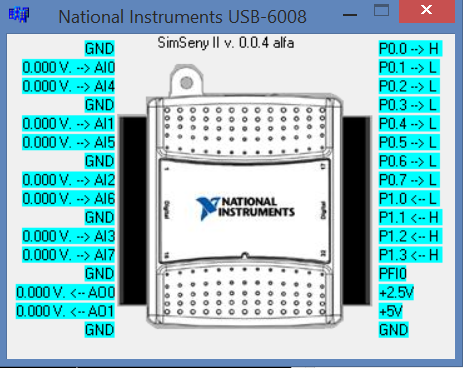
//Activa el ventilador.

1. **void \_\_fastcall** ScrollBarElectroValvulaChange(TObject \*Sender);// Realiza la escritura analógica de la electroválvula.
2. **void \_\_fastcall** ScrollBarVentilacionChange(TObject \*Sender);// Realiza la escritura analógica en la el ventilador.
3. **private**: *// User declarations*
4. **public**: *// User declarations*
5. **\_\_fastcall** TVPrincipal(TComponent\* Owner);
6. };
7. *//---------------------------------------------------------*
8. **extern** PACKAGE TVPrincipal \*VPrincipal;
9. *//---------------------------------------------------------*
10. #endif

**5. Capturas de pantallas y explicación de la aplicación desarrollada**

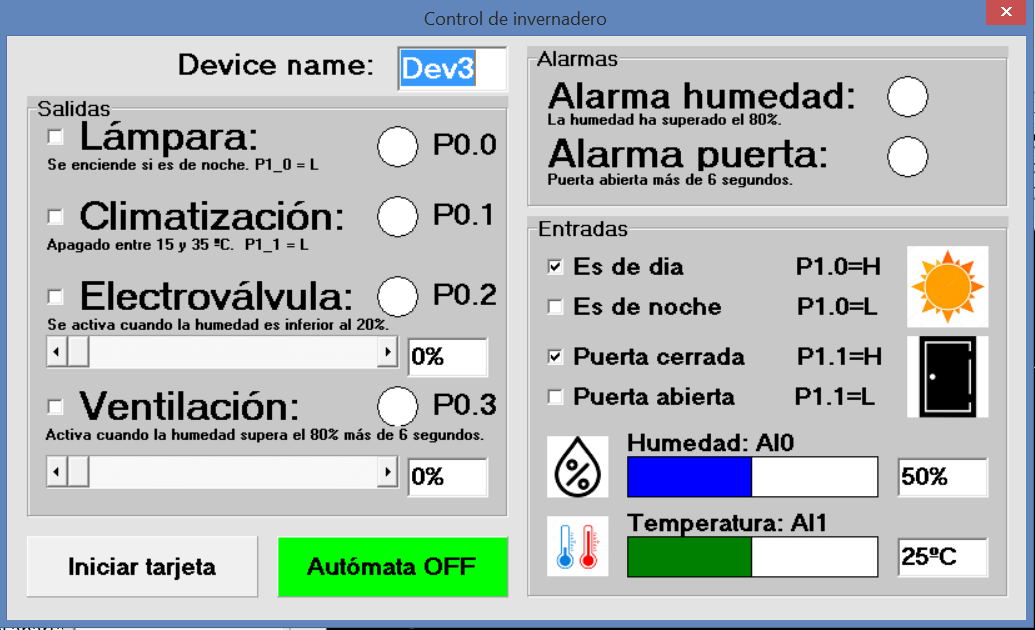
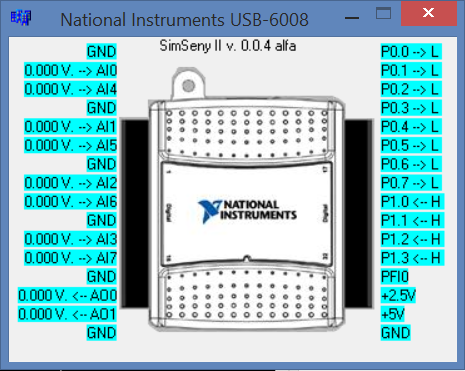


Para iniciar sistema tenemos que empezar pulsando el botón “iniciar tarjeta” este botón pone en funcionamiento la tarjeta de adquisición de datos, seguidamente debemos de pulsar el botón “Autómata On” el cual va iniciar el proceso de control del invernadero. Para saber si se ha realizado bien el inicio debemos visualizar que el botón “Autómata ON”, va alternando su relleno con los colores verde y amarillo además de cambiar su caption a “Autómata OFF”.

Seguidamente y para la visualización del control se hacen los cambios de estados digitales y analógicos sobre las entradas digitales/analógicas de la tarjeta de adquisición de datos digital.

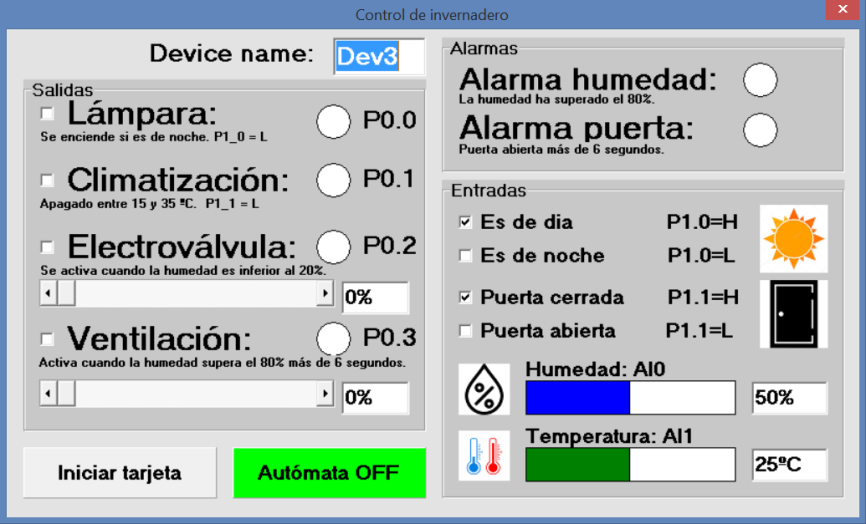
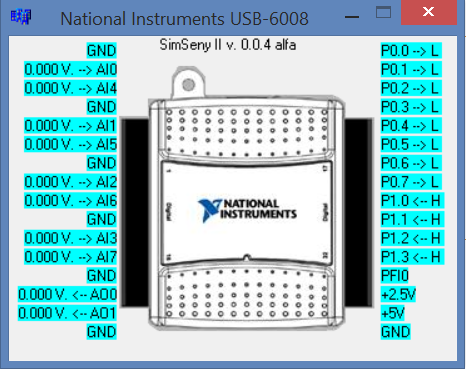
Una vez el proceso de control está activado comprobamos su funcionamiento, para ello colocamos la entrada (P 0.1) a nivel bajo (0), esto significa que en el exterior del invernadero hay oscuridad, por lo tanto en el panel de control se observa que en el apartado de entradas el valor del sensor LDR (P1.0) está a nivel bajo por lo tanto checkbox “Es de noche “está chequeado y aparece la imagen de la luna.

En el apartado de salidas observamos que el checkbox de “lámpara” está activado y además el Shape circulo se rellena de color rojo indicando que la salida (P 0.0) está activa (nivel alto 1).



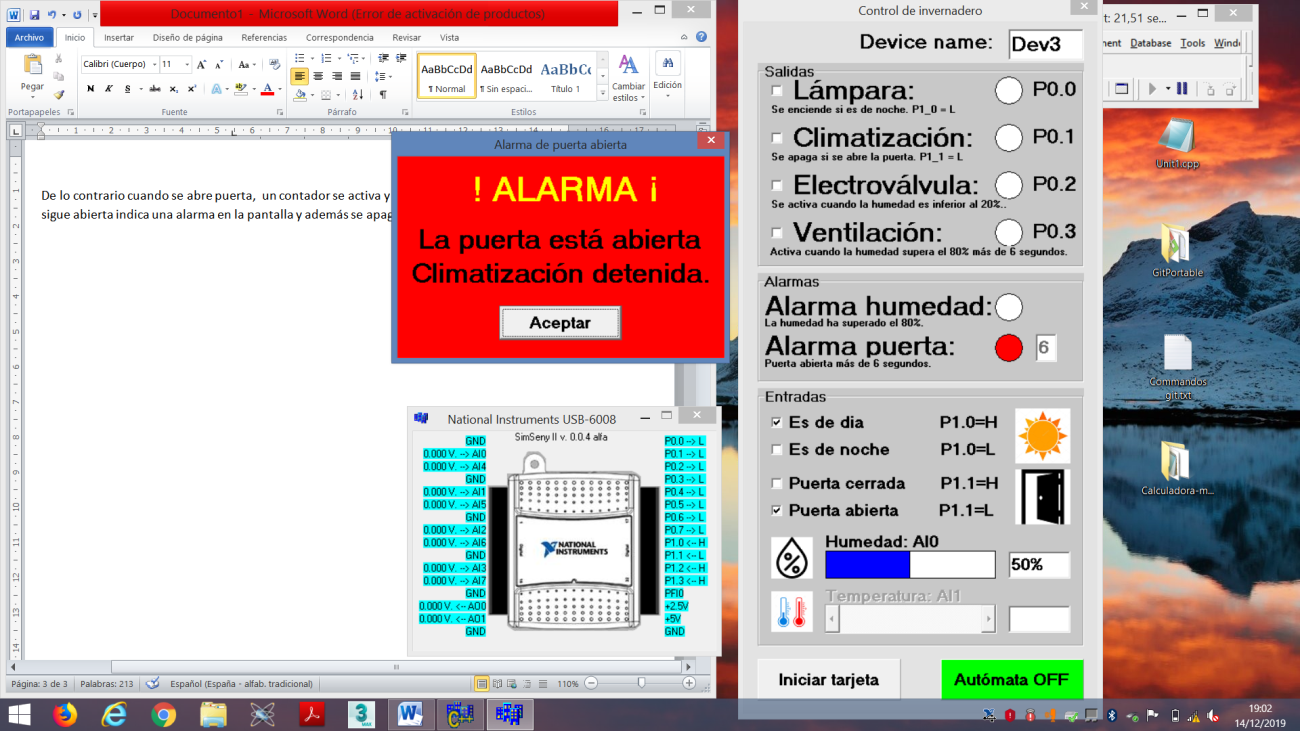
Por el contrario la lámpara se apaga cuando el sensor LDR (P 0.1) lo colocamos nivel alto (1), esto significa que en el exterior del invernadero hay luz, por lo tanto en el panel de control se observa en el apartado de entradas el valor del sensor LDR (P1.0) está a nivel alto por lo tanto checkbox “Es de dia “está chequeado y aparece la imagen de sol.

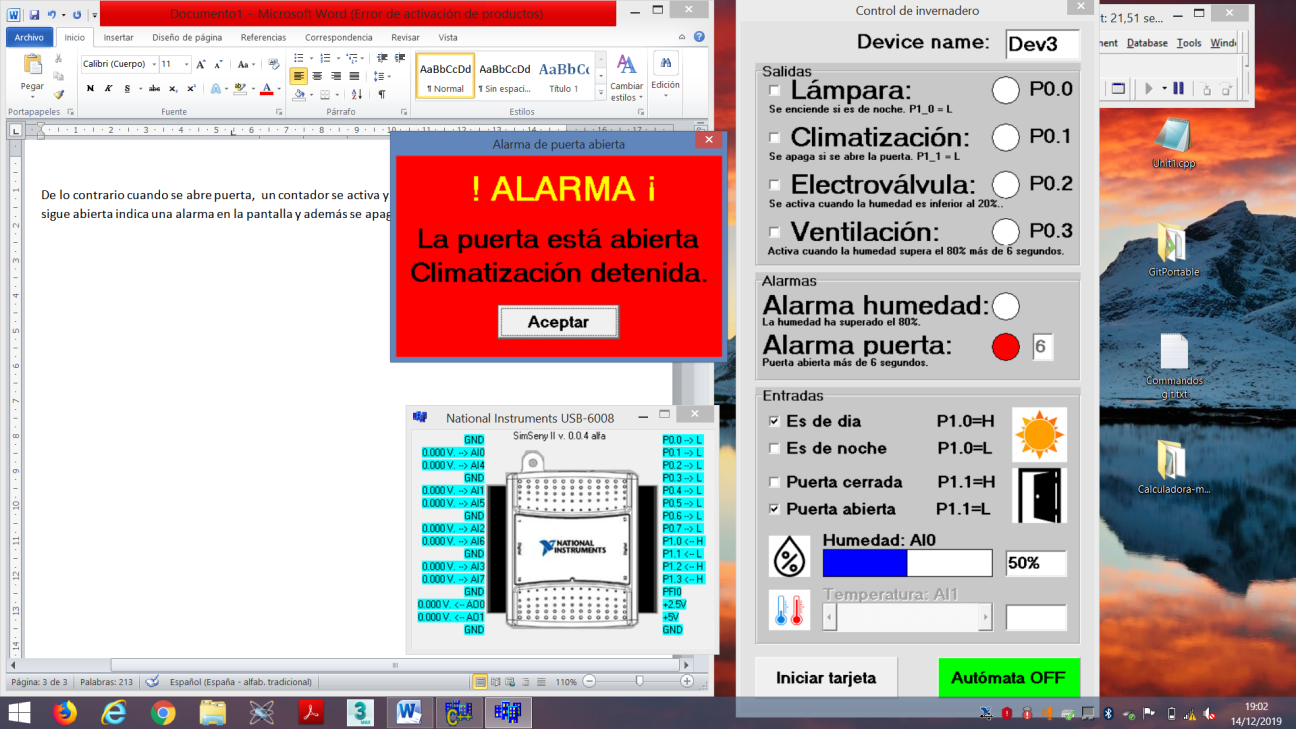
En el apartado de salidas observamos que el checkbox de “lámpara” está deschequeado y además el Shape circulo se colorea de color blanco indicando que la salida (P 0.0) está inactiva (nivel bajo 0).



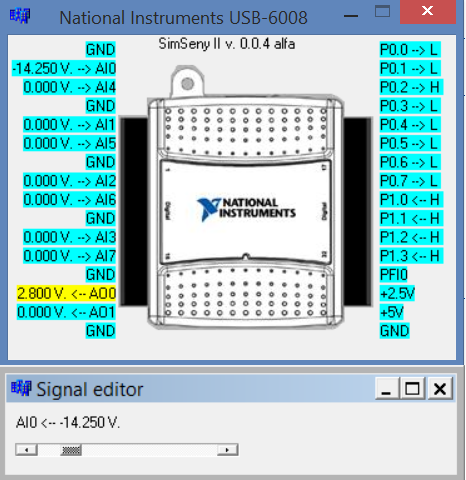
Cuando el sensor de la puerta (P 1.1) está detectando que la puerta está cerrada nivel alto (1). Observamos en el panel de control que en el apartado de entradas, el checkbox “puerta cerrada” (P1.1) está marcado y vemos el dibujo de una puerta cerrada.

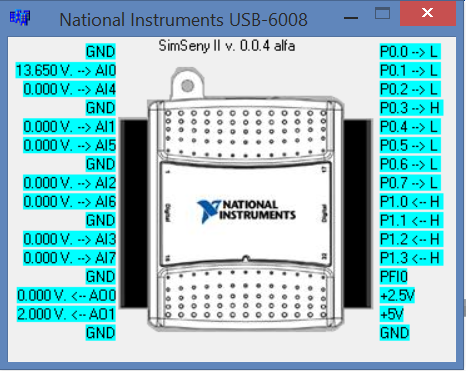
Ahora colocamos el sensor de la puerta (P 1.1) a nivel bajo (0), entonces un contador se activa y se visualiza , si transcurridos 6 segundos la puerta sigue abierta indica una alarma en la pantalla. En este caso en el panel de control, en su apartado de entradas vemos que el checkbox de puerta abierta está marcado y aparece una imagen de una puerta abierta. En el apartado de salidas vemos que no hay nada marcado debido a la alarma producida. En el apartado de alarmas indica que la puerta está abierta con un shape círculo que va alternado su color de relleno rojo-blanco con la representación del temporizador.

****

****

Para eliminar la alarma tan solo hay que apretar el botón de aceptar en la ventana que ha salido en la pantalla y modificar el estado del sensor de la puerta (P 1.1), a estado alto (1). Ahora el sistema se encuentra en la situación descrita anteriormente puerta cerrada.

El siguiente es un control de humedad con un sensor de humedad (AI 0) que trabaja de forma analógica (múltiples estados), si el sensor lee una humedad demasiado baja (Inferior al 20%), la salida (P 0.2) se activa, nivel alto (1).

Para realizar esta acción hay que mover la entrada (AI 0) a su valor inferior al 20%, en el panel de control en su apartado de entradas vemos que el shape rectangular esta coloreado hasta ahora de azul se ha ido reduciendo el relleno proporcionalmente a la variación de humedad impuesta y una vez llegados al límite del 20% cambia de color de relleno a amarillo indicando una anomalía y reduciéndose proporcionalmente con la humedad hasta 0%. En el recuadro contiguo indica el valor del tanto por ciento de humedad que hay en el interior del invernadero. En el apartado de salidas vemos que el checkbox de la electroválvula (P0.2) está marcado y el shape circulo esta relleno de color rojo. Además el control de la válvula es analógica representado por el scrollbar situado en la parte inferior del checkbox electroválvula, en se visualiza la apertura de la electroválvula (AO 0), la apertura comienza a partir de una humedad del 19.9% y va aumentado su apertura a raíz que la humedad siga disminuyendo, coincidiendo en una apertura el 100% con una humedad del 0%. En el recuadro contiguo indica el valor del tanto por ciento de apertura de la electroválvula.

Por otro lado si el sensor de humedad lee una humedad alta (superior al 80%), un contador se activa y si transcurridos 6 segundos sigue por encima del 80% indica una alarma en el cuadro de control.

En el apartado de alarmas en vemos el shape circulo de la alarma de humedad relleno de color rojo-blanco alternado con el recuadro del contador contiguo.

Para realizar esta acción hay que mover la entrada (AI 0) a su valor inferior al 80%, en el panel de control en su apartado de entradas vemos que el shape rectangular esta coloreado hasta ahora de azul se ha ido incrementando el relleno proporcionalmente a la variación de humedad impuesta y una vez llegados al límite del 80% cambia de color de relleno a rojo indicando una anomalía e incrementándose proporcionalmente con la humedad hasta 100%.

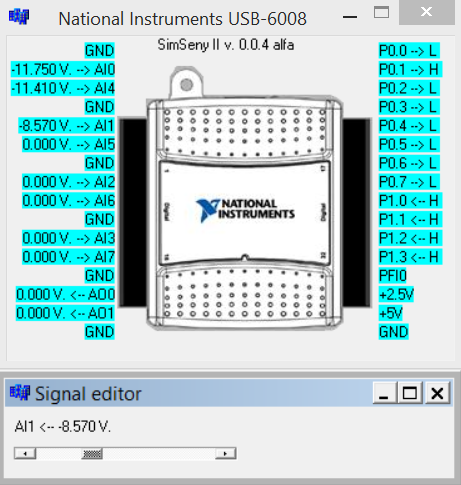
En el recuadro contiguo indica el valor del tanto por ciento de humedad que hay en el interior del invernadero.

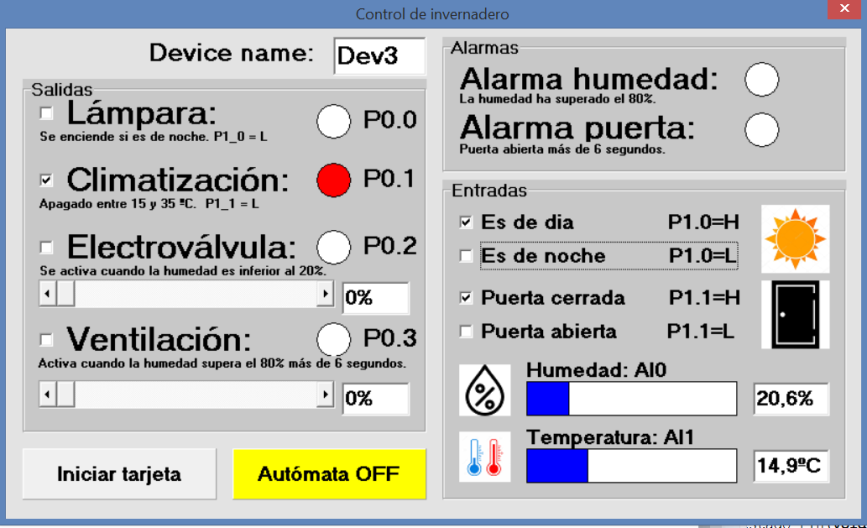
En el apartado de salidas vemos que el checkbox de la ventilación (P0.3) está marcado y el shape circulo esta relleno de color rojo. Además el control del ventilador es analógico y esta representado por el scrollbar situado en la parte inferior del checkbox ventilador, en el visualiza la tensión suministrada al ventilador (AO 1), la tensión comienza a partir de una humedad del 80.02% y va aumentado su valor a raíz que la humedad siga aumentando, coincidiendo en una tensión del 100% con una humedad del 100%.

En el recuadro contiguo indica el valor del tanto por ciento de tensión del ventilador.

Para eliminar la alarma tan solo hay que poner un valor intermedio en la entrada de latarjeta (AI 0).

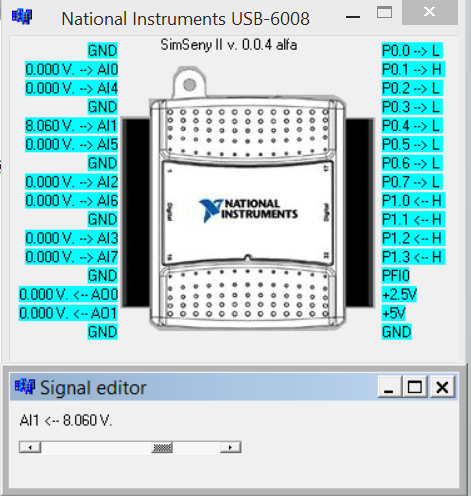
El último control instalado es el de temperatura del interior del invernadero, para este control se ha usado un sensor de temperatura analógico (AI 1).

El funcionamiento es el siguiente, si el sensor lee una temperatura demasiado baja (Inferior a 15˚), la salida (P 0.1) se activa, nivel alto (1).



Para realizar esta acción hay que mover la entrada (AI 1) a su valor inferior al 14.8˚, en el panel de control en su apartado de entradas vemos que el shape rectangular esta coloreado hasta ahora de color verde se ha ido reduciendo el relleno proporcionalmente a la variación de temperatura impuesta y una vez llegados al límite del 14.8˚, cambia de color de relleno a azul indicando una anomalía y reduciéndose proporcionalmente con la temperatura hasta 0˚. En el recuadro contiguo indica el valor de temperatura en valor de grados Celsius que hay en el interior del invernadero.

En el apartado de salidas vemos que el checkbox de la climatización (P0.1) está chequeado y el shape circulo esta relleno de color rojo indicando que la salida esta a nivel alto (1).

Si por el contrario, el sensor lee una temperatura demasiado alta(Superior a 35˚), la salida (P0.1) se activa, nivel alto (1).

Para realizar esta acción hay que mover la entrada (AI 1) a un valor superior a 35˚, en el panel de control en su apartado de entradas vemos que el shape rectangular esta coloreado hasta ahora de color verde se ha ido aumentando el relleno proporcionalmente a la variación de temperatura impuesta y una vez llegados al límite de 35˚, cambia de color de relleno a rojo indicando una anomalía y aumentandose proporcionalmente con la temperatura hasta 50˚. En el recuadro contiguo indica el valor de temperatura en valor de grados Celsius, que hay en el interior del invernadero.

En el apartado de salidas vemos que el checkbox de la climatización (P0.1) está chequeado y el shape circulo esta relleno de color rojo indicando que la salida esta a nivel alto (1)

