



V A N T A G E™

CAMBIAR EL COLOR DE LA RETROILUMINACIÓN DE LAS **BOTONERAS EASYTOUCH SEGÚN LA FUNCIÓN QUE CONTROLAN**

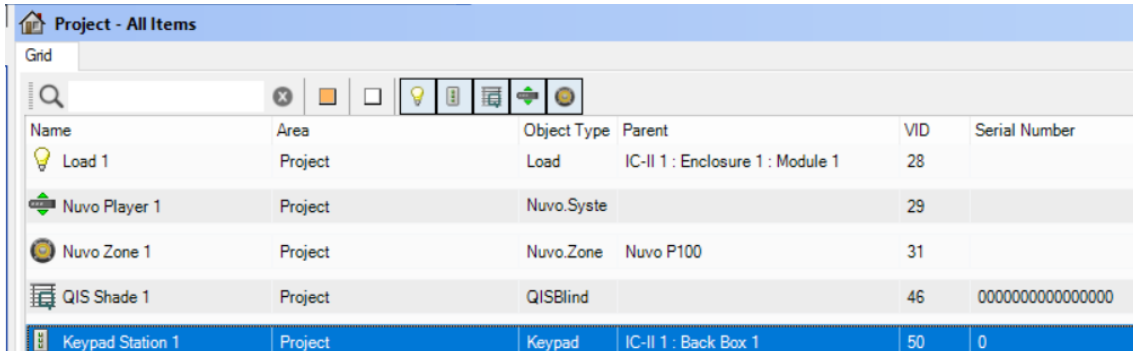


BOTONERA
EASYTOUCH II

Abril de 2020

ASISTENCIA TÉCNICA DE BTICINO DE MÉXICO

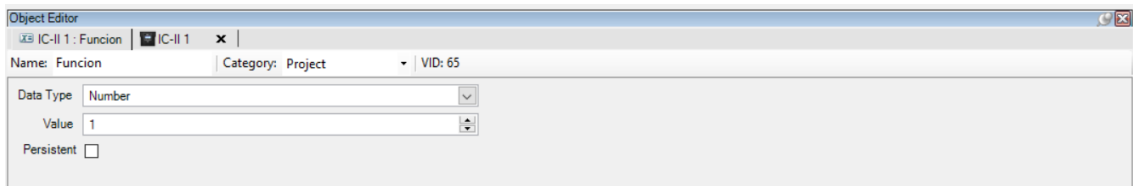
Paso 1: Para este ejemplo, supongamos que tenemos tres funciones distintas que queremos controlar en nuestra botonera Vantage: Iluminación, Persianas y Audio. Comenzamos por agregar una carga a nuestro módulo, una persiana Qmotion y un P100 de Nuvo a nuestro proyecto para simular las tres distintas funciones que vamos a simular:



Name	Area	Object Type	Parent	VID	Serial Number
Load 1	Project	Load	IC-II 1 : Enclosure 1 : Module 1	28	
Nuvo Player 1	Project	Nuvo.System		29	
Nuvo Zone 1	Project	Nuvo.Zone	Nuvo P100	31	
QIS Shade 1	Project	QISBlind		46	0000000000000000
Keypad Station 1	Project	Keypad	IC-II 1 : Back Box 1	50	0

Paso 2: El siguiente paso, sería pasarnos a la sección del Programming View, y ahí vamos a crear una variable numérica ya que queremos controlar tres funciones distintas. En caso de que quisieramos controlar solamente dos, también sería válido el uso de una variable booleana. El atajo del teclado para agregar una variable es Ctrl+G.

La variable que declaremos, para este ejemplo la llamaré “Función” y la voy a inicializar en el valor 1. Además voy a desmarcar la casilla “Persistent”. La casilla “Persistent” se ocupa cuando queremos que la variable conserve su valor en caso de que exista un fallo de energía y luego cuando se reestablezca, la variable conserve el valor que tenía antes de que ocurriera dicho evento. Si desmarcamos la casilla, la variable volverá a tener el valor con el cuál inicializamos la variable en caso de falla de energía.



Object Editor
 IC-II 1 : Funcion | IC-II 1 x
 Name: Funcion | Category: Project | VID: 65
 Data Type: Number
 Value: 1
 Persistent:

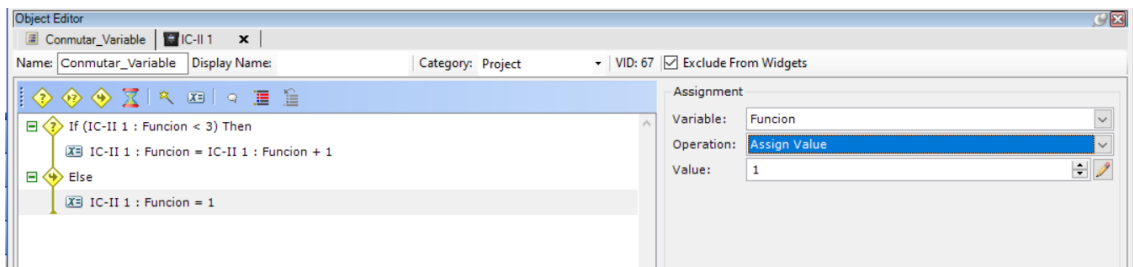
Paso 3: Una vez que tenemos declarada la variable, el siguiente paso sería seleccionar un botón de la botonera Easytouch para que nos sirva como la función de “Shift” y que podamos conmutar entre las tres distintas opciones de control.

Podemos seleccionar ya sea un botón para que tenga esta función exclusiva de conmutar entre las tres distintas opciones, o bien asignar uno que sin que pierda ninguna funcionalidad de control previamente programada, nos sirva además para conmutar entre las opciones.

Para este ejemplo optaremos por esa segunda opción. Para ello vamos a utilizar el procedimiento Press & Hold, que lo que hace es ejecutar dos tareas distintas, una cuando se realiza una presión rápida del botón y otra distinta cuando presionamos y mantenemos presionado el botón.



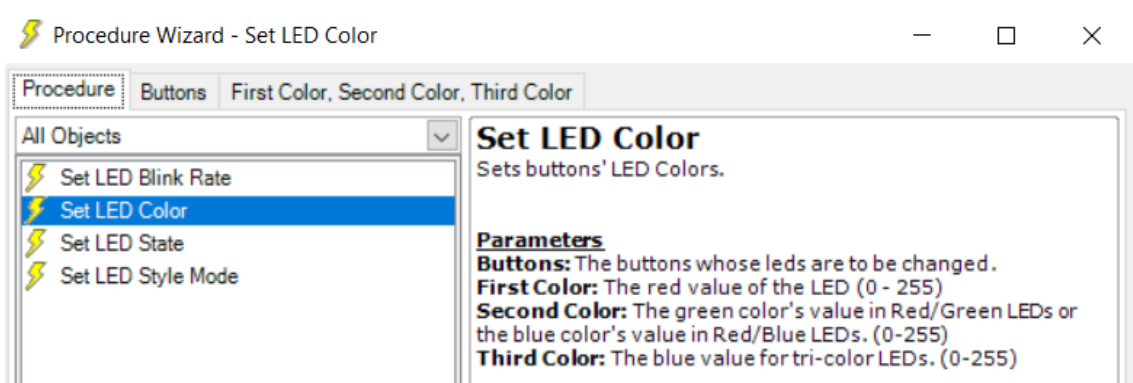
La idea es hacer que cuando mantengamos presionado el botón, este incremente su valor hasta llegar a 3, y cuando llegemos a tres, regrese al valor de 1. Esto lo logramos programando una tarea que tenga las siguientes condiciones:



Paso 5: Vez que tenemos lista la tarea “Conmutar_variable” procedemos a crear una tarea que controle las tres distintas funciones según el valor que tenga en ese momento la variable de acuerdo con el valor que tenga la variable en ese momento. Además, sería deseable dotar al usuario con una señal visual que le permita identificar qué es lo que va a estar controlando. En este caso lo que vamos a hacer será cambiar el color de la botonera para que el usuario para que sea fácil para él identificar la función que va a controlar de acuerdo a la siguiente tabla:

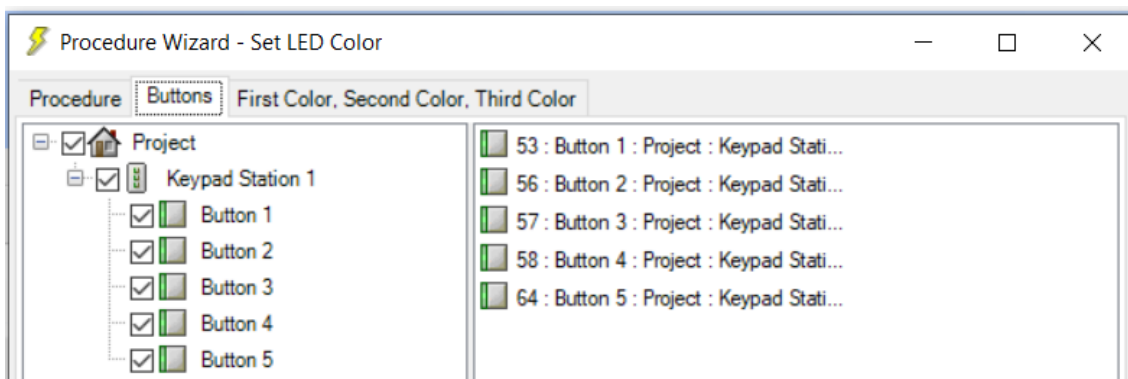
Valor de la variable	Función que vamos a controlar	Color de la botonera
1	Iluminación	Rojo
2	Persianas	Verde
3	Audio	Azul

Para cambiar el color de la iluminación de la botonera de forma fácil, podemos utilizar el procedimiento “Set LED Color”.

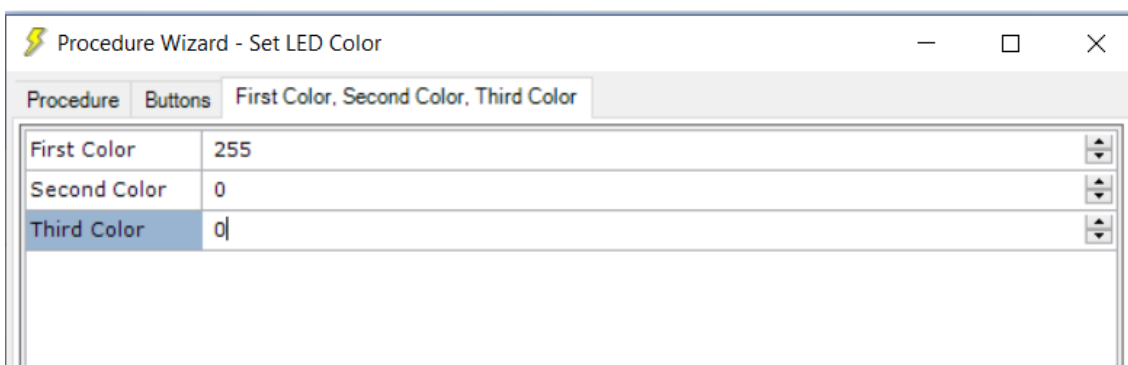


Para poder hacer uso de la función Set LED Color, básicamente requerimos de dos parámetros:

1. Los botones de los que queremos cambiar el color de la botonera.



2. El valor en formato RGB del color que queremos que tenga la retroiluminación:



Debido a que RGB significa (Red Green Blue) que son los tres colores primarios de la luz, tenemos que asignar un valor de entre 0 y 255 en cada valor para ir formando las distintas combinaciones de colores. En este caso para obtener los tres valores primarios, los valores que le tenemos que dar a tabla son los siguientes:

Rojo	
First Color	255
Second Color	0
Third Color	0

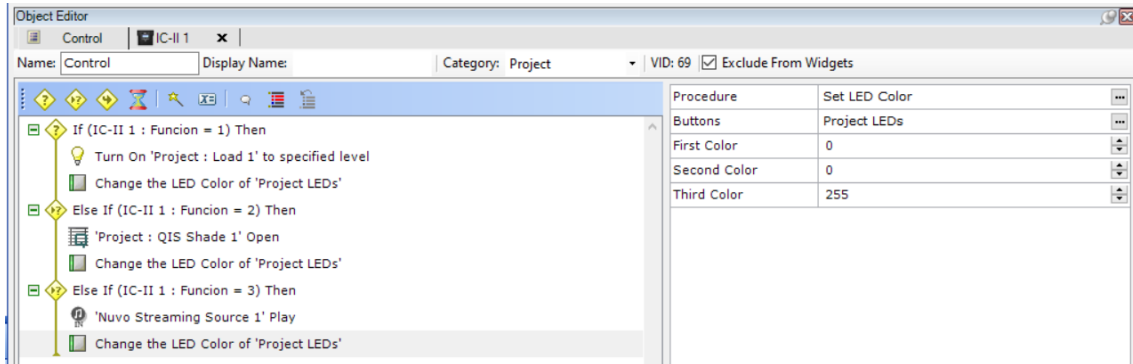
Verde	
First Color	0
Second Color	255
Third Color	0

Azul	
First Color	0
Second Color	0
Third Color	255



Para obtener los valores que debemos darle a esa tabla para conseguir cualquier otro color, se recomienda utilizar páginas gratuitas como la siguiente:

<https://htmlcolorcodes.com/es/>

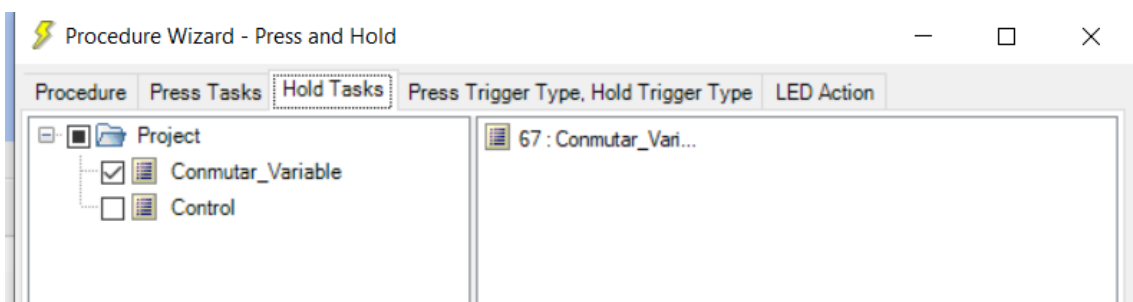
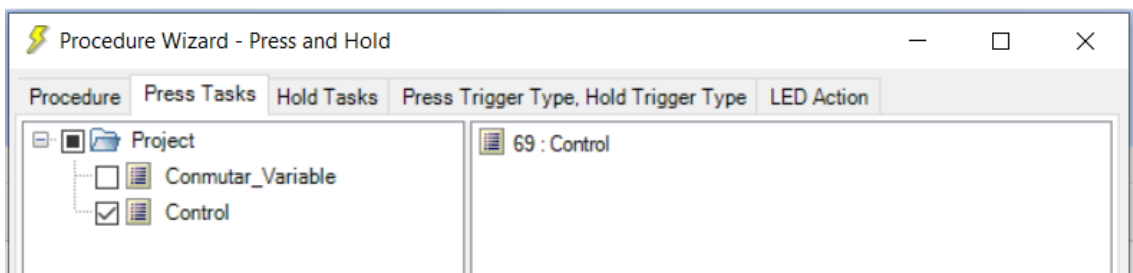


Como puede observarse en la imagen anterior, de acuerdo con el valor que tenga en ese momento la variable, va a ser la función que va a realizar ya que sea encender un circuito si la variable vale 1, abrir una persiana si la variable vale 2 o darle comenzar a reproducir música en una zona de Nuvo si la variable vale 3.

Debido a que el cambio del color de la retroiluminación es una consecuencia directa del cambio de valor de la variable que representa solamente una indicación visual para ayudar a identificar al usuario qué función va a controlar, los colores de la botonera se pueden cambiar ya sea en la tarea “conmutar_variable” o bien en la tarea “control”.

Paso 6: Ahora, es necesario utilizar la función “Press & Hold” para hacer que las dos tareas se puedan ejecutar desde un mismo botón en función del tipo de presión que ejerza el usuario sobre el botón ya sea una pulsación corta o sostenida.

Es importante seleccionar como Press task la tarea control y como Hold task la tarea de “conmutar_variable”. Si se seleccionan al revés, la tarea de control va a ser ejecutada con un pulso sostenido:



Los valores para Press Trigger Type, Hold Trigger Type son opcionales y por lo general se dejan en el valor que viene por defecto "None".

Paso 7: El último paso sería asignar a un botón de la botonera la tarea "Press&Hold" para que ya tengamos todas las funcionalidades que acabamos de programar.

