



HMI  
SYSTEMS

# CAN Create User's Manual

Rev 1

1-13-2012

# CAN Create User's Manual

## Index

---

### **Chapter 1 – Installation pg 3**

- 1.1 Windows XP/Windows 7
- 1.2 Set up Image Library
- 1.3 Installing Program to Device
- 1.4 Necessary Batch Files

### **Chapter 2 – Interface pg 7**

- 2.1 CAN Create Workspace
- 2.2 File Toolbar
- 2.3 Quick-Access Toolbar
- 2.4 Tools Sidebar
- 2.5 Part Descriptions

### **Chapter 3 - The Basics pg 26**

- 3.1 Printing to Display
- 3.2 Using Bitmaps
- 3.3 Using Touch
- 3.4 Creating Pages
- 3.5 Using Text
- 3.6 Using a Counter and Math Parts
- 3.7 Running the Simulator
- 3.8 Result

### **Chapter 4 - Basic Animation pg 35**

- 4.1 Blinking
- 4.2 Image Slideshow

### **Chapter 5- Labels and Inputs pg 39**

- 5.1 Labeling
- 5.2 Extension: More Labeling
- 5.3 Multi-Function Input (Short/Long Presses)
- 5.4 Result

**Chapter 6- Calendar and Time   pg 46**

- 6.1 Calendar Part
- 6.2 Clock Part
- 6.3 Result

**Chapter 7- Advanced Page Navigation   pg 52**

- 7.1 Multiple Page Setup
- 7.2 Using Buffers for Pages
- 7.3 Result

**Chapter 8- CAN Receiver   pg 58**

- 8.1 Using CANRX
- 8.2 Simulation Values
- 8.3 Result
- 8.4 Extension: Execution Uses

**Chapter 9- CAN Transmitter   pg 65**

- 9.1 Using Personal Font
- 9.2 Using CANTX
- 9.3 Result

**Chapter 10- Simulating with Hardware   pg 80**

- 10.1 Using the Message Window
- 10.2 Using a Switch

**Chapter 11- Troubleshooting FAQ   pg 84**

- 11.1 Locating Installation Location
- 11.2 Preventing Accidental Touch
- 11.3 Touch Parts “Not Working”
- 11.4 Device Time is Incorrect
- 11.5 Bitmap Rendering Too Slow
- 11.6 Bitmap Won’t Disappear
- 11.7 Why are Bitmaps Renamed
- 11.8 Bitmap Image Support
- 11.9 Tutorial Images not Displayed
- 11.10 New Library Images not Listed
- 11.11 Design not Working Correctly

# Chapter 1 - Installation

## 1.1 Windows XP/Windows 7

Once HMI Systems has provided your CAN Create zip file, follow the steps below to install the software onto your computer.

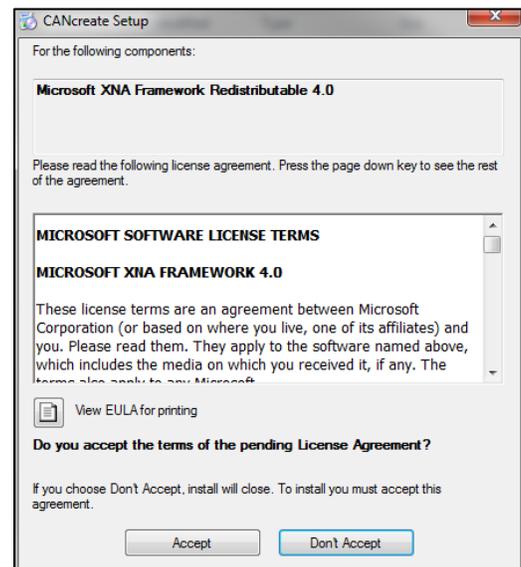
1. First locate your zip file. Then extract the file to a folder using a standard extracting tool (Windows extractor, WinRAR, WinZip, etc) by right clicking and selecting 'extract'. Given the option, extract to folder on the desktop.

2. Open the newly created folder and there should be three files (seen to the right).



3. Double click the 'setup' execution file to begin installation.

4. The CANcreate Setup window will open up on your screen asking to accept the license agreement for Microsoft XNA Framework. Click 'Accept' to continue.



5. Setup will now download Microsoft XNA Framework and install it. This can take a few minutes.

6. After setup is done installing, it is possible a 'Publisher cannot be verified' window will appear, click install anyway to continue.

7. The installation is now complete; the CANcreate interface window should open on your screen.  
-If not, use CANcreate execution file in the extract folder to run



8. To locate the installed folder, follow the instructions under the "Set Up Image Library" section.

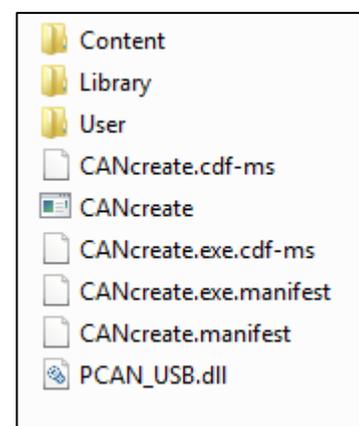
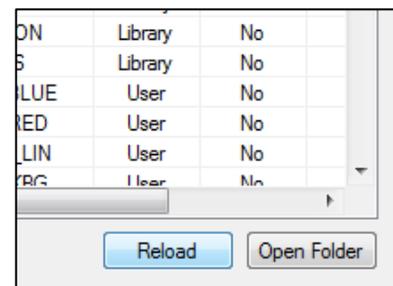
## 1.2 Set Up Image Library

---

### Windows XP/Windows 7:

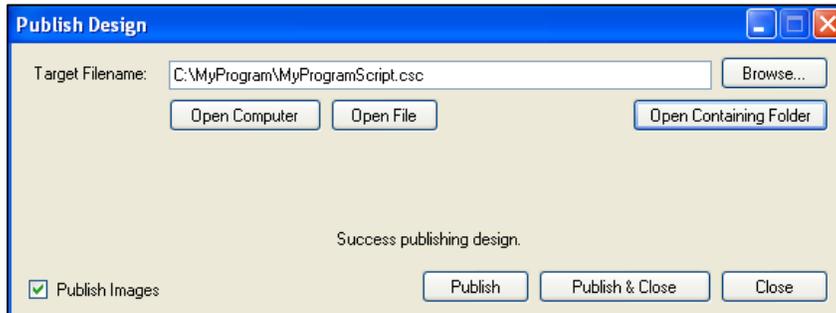
With Windows 7, run the CANcreate software. If it is not open, find the original extracted folder (ex: User\Name\Desktop\CANcreate if on desktop). Open the folder and double-click the “CANcreate” application run file. The software should open up. Next, go to tools and click “Open Image Library”. Find the “Open Folder” button and click it. The folder that opens up is the application folder. (Can also go to Help->About and use the Open Application Folder button to locate the Image Folder) You can choose to leave the folder in this location or copy all the contents into a newly created folder somewhere more preferable.

Navigate to \User folder within your CANcreate folder and copy the “Manual” folder to this location. Any images or folders put into this \User folder will be accessible within the CANcreate program. When setting BITMAP or BACKGROUND properties, the folder can be set to User\Manual to access asset images.



## 1.3 Installing Program to Device

---

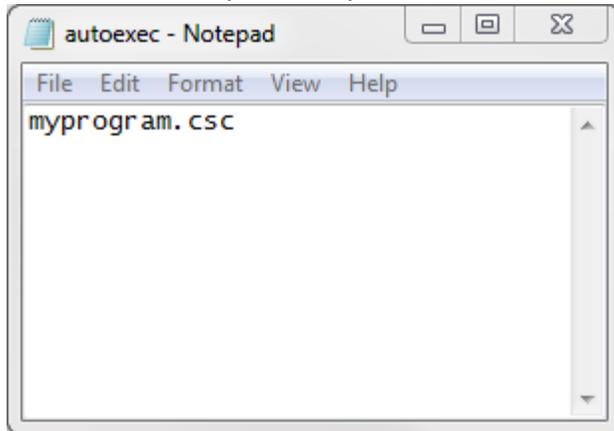


1. Once completed with a project, go to Publish and Select Publish (or press [F5] )
2. A window titled “Publish Design” will open. In the Target Filename, use browse to select a folder that will be copied onto the microSD card. And name your program's script.  
**Example:** If new folder is MyProgram and name for script is MyProgramScript:  
Target Filename: C:\MyProgram\MyProgramScript.csc
3. Make sure Publish Images box is checked and click Publish. When completed, the message “Success publishing design.” will be displayed.  
**Note:** Can use Open File to view the contents of the script file or Open Containing Folder to see contents of the folder.
4. Next, connect your microSD card to the computer and navigate to My Computer. Locate the microSD Drive, right-click and select Format. A window will pop up, change File System drop-down menu to FAT32. Create a volume name, such as “HMILCD”. Click Start.
5. Copy the entire newly created folder contents onto the microSD card. Then two batch files need to be added onto the card. Open a text file, type in the name of your program script with the “.csc” ending. Save the file onto the microSD as “autoexec.bat”. Create a second text file, type “clock”, then on the line below it type the name of your program script with the “.csc” ending. Save this file onto the microSD as “diag.bat”. When completed, eject the microSD card and it is ready to run on your touch display.

## 1.4 Necessary Batch Files

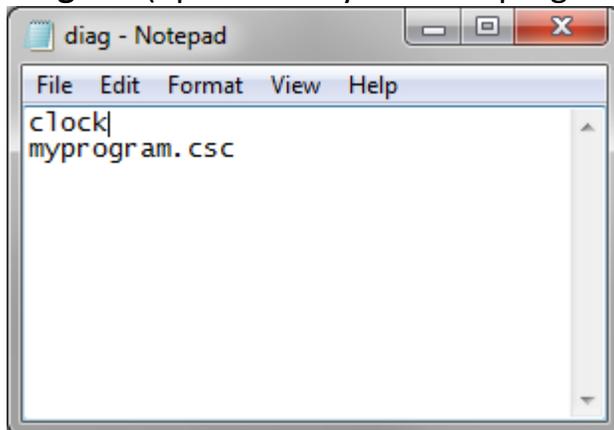
---

### Autoexec.bat (needed)



A screenshot of a Notepad window titled "autoexec - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text area contains the text "myprogram.csc".

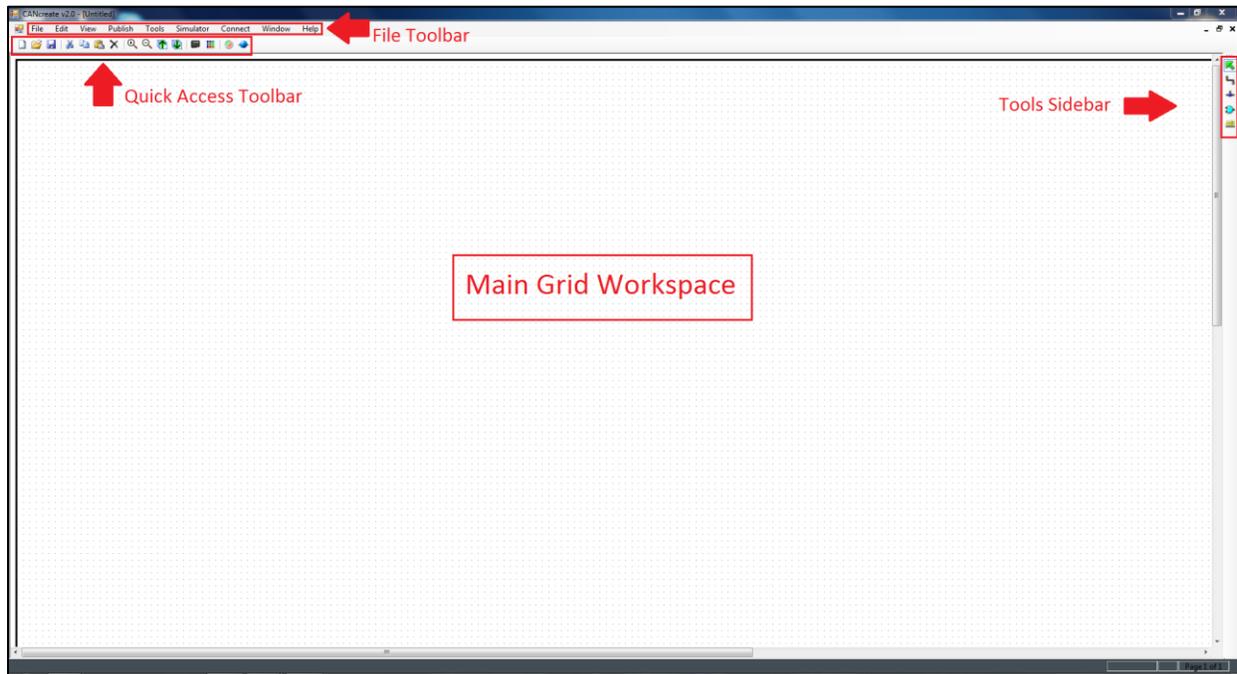
### Diag.bat (optional-only used for programs involving time)



A screenshot of a Notepad window titled "diag - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text area contains the text "clock|" on the first line and "myprogram.csc" on the second line.

# Chapter 2 - Interface

## 2.1 CAN Create Workspace



Once installed, the CAN Create workspace (above) will be shown. The main grid area will be where the wires and parts will be combined together to program your device. There are 3 main sections to the interface: the File toolbar, the quick-access toolbar, and the tools sidebar.

### 2.2-2.9 – File Toolbar

### 2.10 –Quick-Access Toolbar

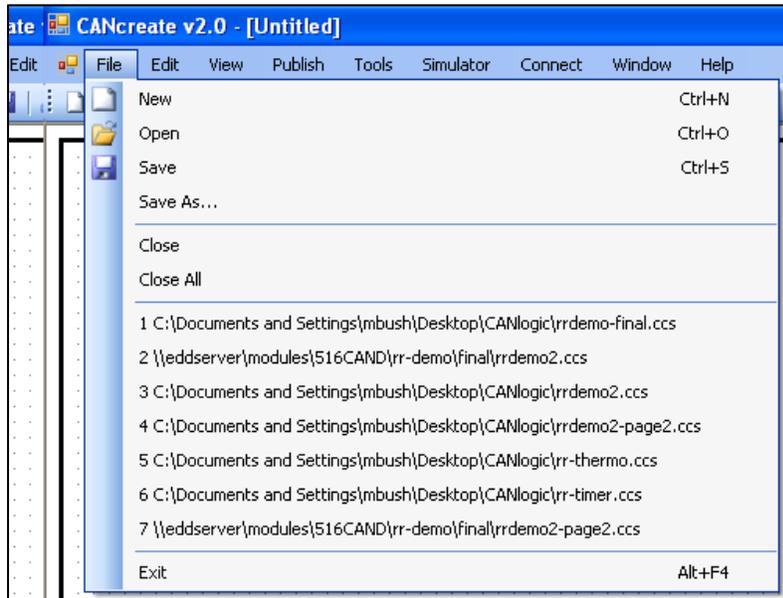
### 2.11 –Tools Sidebar

## 2.2 File Toolbar-File

---



Under the **File menu**, options include:



**New** – Create a new blank CANcreate workspace

**Open** – Open a saved CANcreate project from a folder located on your computer

**Save** – Save the current CANcreate project

**Save As** – Save the current project in a specified location

**Close** – Close the current active project workspace

**Close All** – Close all active project workspaces

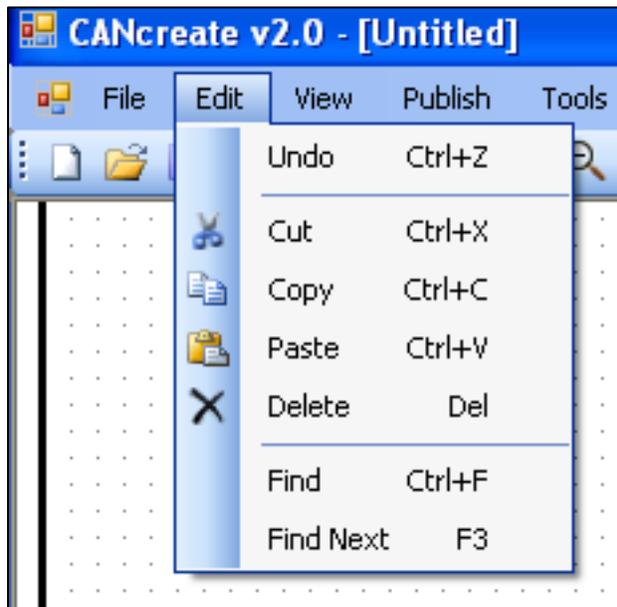
**List of previously opened documents** – Re-open recent files

**Exit** – Close CANcreate application

## 2.3 File Toolbar-Edit

---

Under the **Edit menu**, options include:



**Undo** – Undo a previous action

**Cut** – Copy and delete a selected object

**Copy** – Copy a selected object

**Paste** – Paste a selected object

**Delete** – Delete a selected object

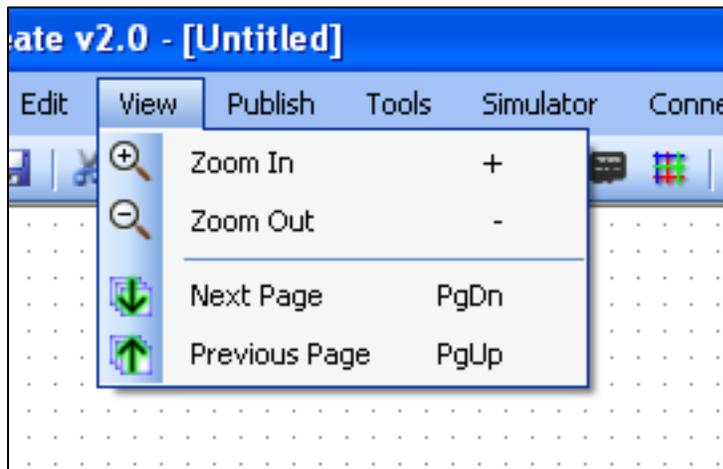
**Find** – Find a certain part by typing in label or using the drop down menu

**Find Next** – Use to find the next part of the same type after using the find option

## 2.4 File Toolbar-View

---

Under the **View menu**, options include:



To navigate and get an overview of the workspace use zoom

**-Zoom In (+)**

**-Zoom Out (-)**

When designing your GUI, for organization, parts can be created on multiple pages.

**Note:** Each page isn't its own project, parts will interact with each other, and labeled wires will connect across multiple pages.

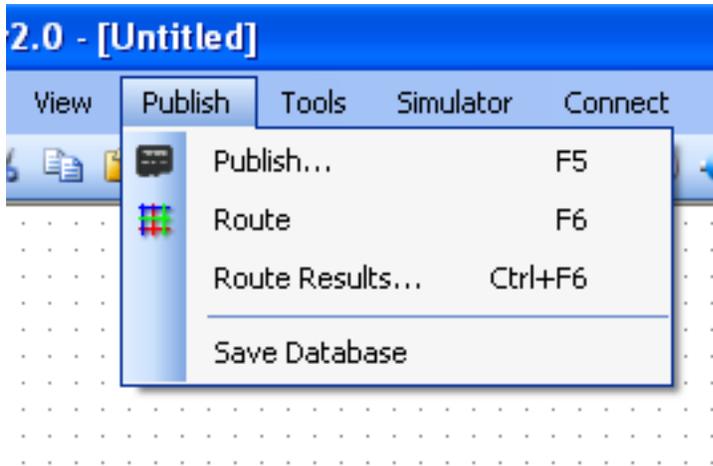
**-Next Page (Page Down)**

**-Previous Page (Page Up)**

## 2.5 File Toolbar-Publish

---

Under the **Publish menu**, options include:



**Publish**- compile the design and formats the graphics into the structure that the display can read (script file is created)

**Route**- compiles the design (physical)

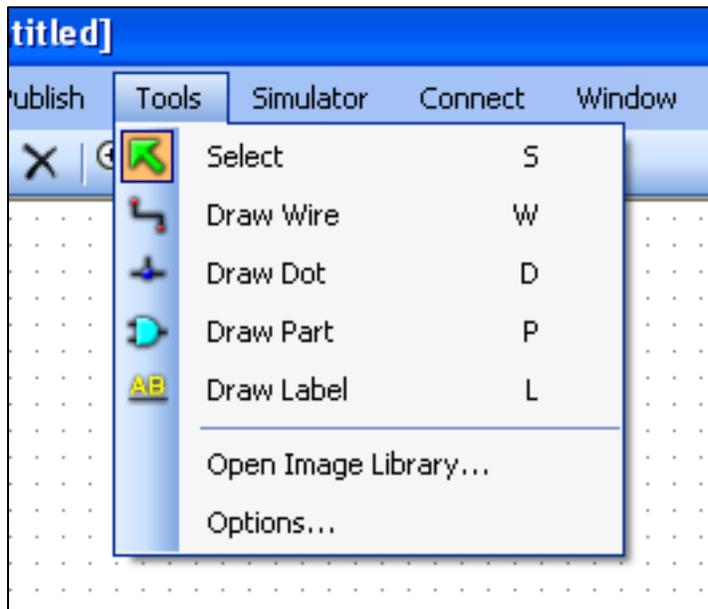
**Route Results**- gives warnings and errors after compiles

**Save Database**- Archives the component library (Debug only)

## 2.6 File Toolbar-Tools

---

Under the **Tools menu**, options include:



**Select**- Used to click, drag, and edit parts within the workspace

**Draw Wire**- Left-click to start drawing a wire, left-click to end wire, and left double-click will terminate wire and open wire properties. Also can right-click to stop creating wire without going into properties.

**Draw Dot**- Place an intersection dot on crossing wires when dot doesn't automatically appear

**Draw Part**- Place a building component part into the workspace.

Contains categories:

Constants, Display, Flops, Gates, Math, and Timers

**Draw Label**- One click wire properties instead of double-clicking

**Open Image Library**- Shows a list of images (loaded/unloaded) in /CANlogic/user folder and subfolders

## 2.6 File Toolbar-Tools (Cont.)

---

### Options:

**Show All End Points**- Displays all end points of wires, not just intersection points

**Show All Labels**- Displays the default labels of all wires used

**Show Part References**- Displays, on all parts, the reference code name

**Show Route Order**- Labels parts with numbers representing order of execution

**Show Input Pins**- Highlights the input pins of all parts

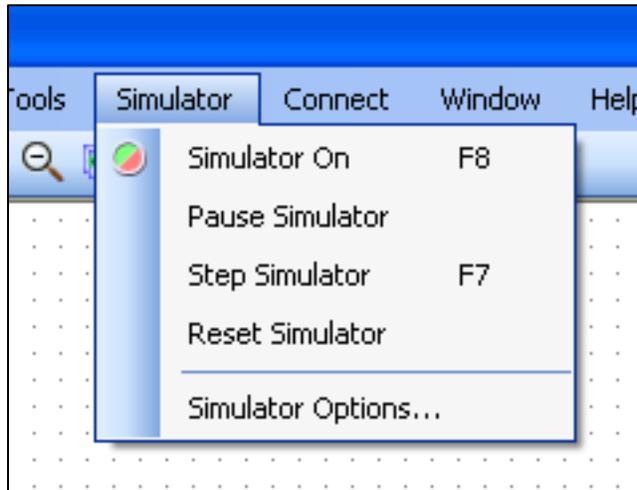
**Auto Save**- Set auto save on or off; Set how often to save

**Max Undo**- Set the maximum number of times for undo-ing actions

## 2.7 File Toolbar-Simulator

---

Under the **Simulator menu**, options include:



**Simulator On (F8)** – Starts the simulator for current program from last run state

(When running, the icon on the quick access toolbar is highlighted)

**Simulator On:** 

**Simulator Off:** 

**Wire Colors during Simulation:**

**Black-** Contains False value 

**Green-** Contains True value 

**Blue-** Contains Integer value 

**Pause Simulator** – Pauses simulator at current point in time

**Step Simulator (F7)** – Runs simulator step by step through program

**Reset Simulator** – Resets the simulator values to original states

**Simulator Options:**

**Update Period:** Default is 100ms, specifies how quickly the simulator updates while running

## 2.7 File Toolbar-Simulator (Cont.)

---

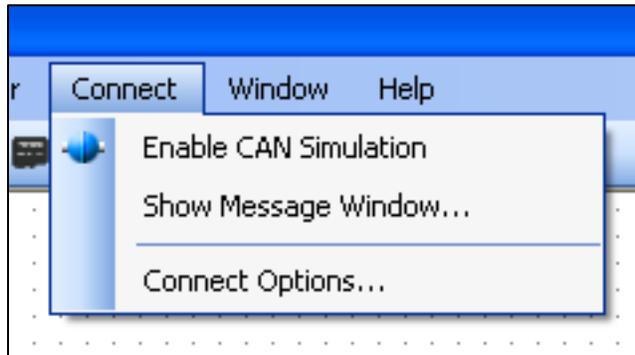
**Steps Per Update:** Default is 1, specifies how many steps are changed in the simulation during each period

**Note:** In order to simulate timer resolutions below 100ms, the update period and steps per update must be changed from default. In general, period should be half the lowest timescale and at least 2 updates per period to capture 50% duty cycle pulses. (Timer objects are affected more than delay objects)

## 2.8 File Toolbar-Connect

---

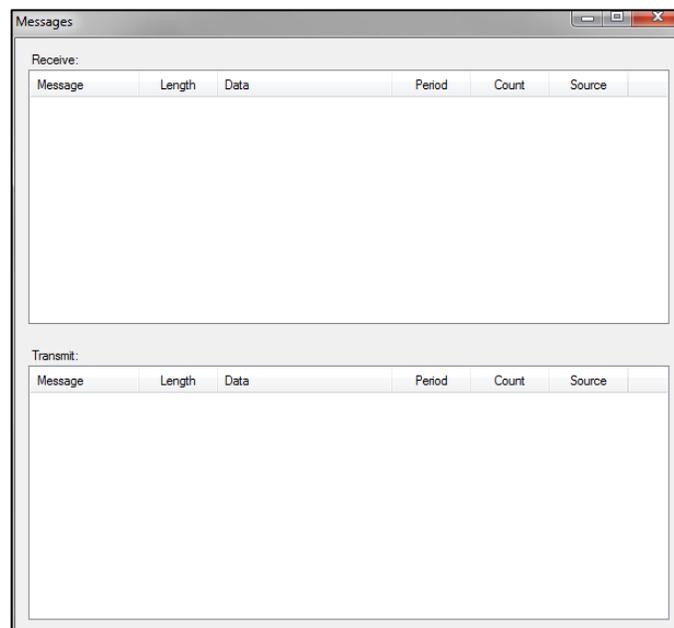
Under the **Connect menu**, options include:



**Enable CAN Simulation**- Use to attach USB from physical device and observe program running in the simulator while running on device. Can be used to change test values and other data manipulation. When running, the icon on the quick access toolbar is highlighted with a blue box.

**Show Message Window**- Shows the data receiving and transmitting messages

-Can right-click and simulate hardware incoming messages



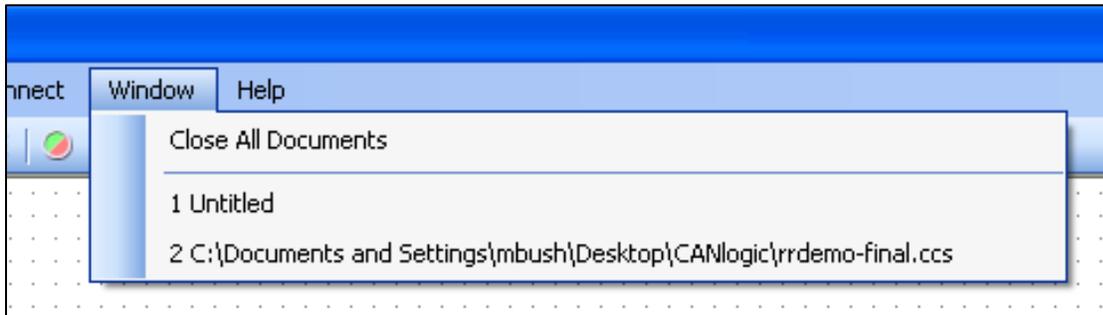
**Connect Options:**

**Connection Type:** Switch between PCAN GridConnect USB and 514UTC USB

## 2.9 File Toolbar-window & Help

---

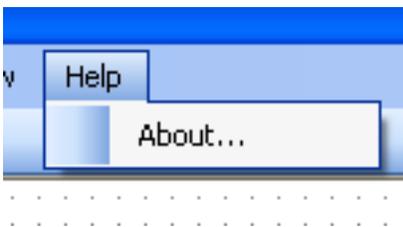
Under the **Window menu**, options include:



**Close All Documents-** Closes all open projects

**List of opened windows-** Gives a list of currently open projects, which can be selected to be the current active project

Under the **Help menu**, options include:



**About-** Version number and copyright information

- Open Application Folder locates the image library

## 2.10 Quick-Access Toolbar



New Blank Project



Open Project



Save Project



Cut



Copy



Paste



Delete



Zoom in



Zoom out



Previous Page



Next page



Publish Project



Route Project



Turn On/Off Simulation



Connect to Device

### Shortcuts:

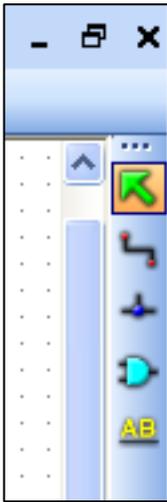
New - [Ctrl]+[N]      Publish - [F5]

Open - [Ctrl]+[O]      Route - [F6]

Save - [Ctrl]+[S]      Simulator On - [F8]

## 2.11 Tools sidebar

---



 Select [S]

 Draw Wire [W]

 Draw Dot [D]

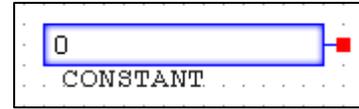
 Draw Part [P]

 Place Label [L]

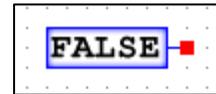
## 2.12 Parts Descriptions

### Constants:

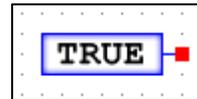
**Constant**- Set a constant variable (Integer Value-Max 32 bit signed)



**False**- Sets connection to Boolean false

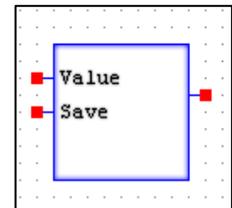


**True**- Sets connection to Boolean true



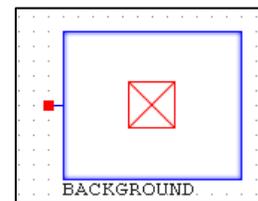
### Disk:

**Memory**- Set, save and restore value to/from memory address on device

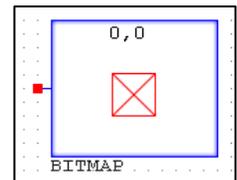


### Display:

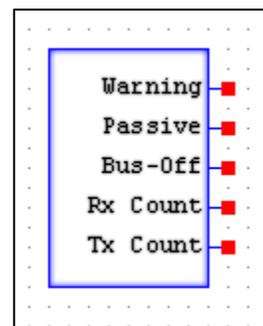
**Background**- Sets a background image on display (320x240 max resolution)



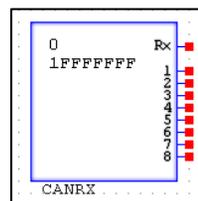
**Bitmap**- Sets image on top of background and is moveable over (X,Y) coordinates



**CANError**- Use to debug and error check

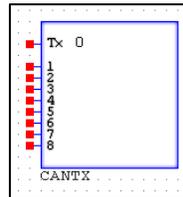


**CANRX**- CAN system receiver

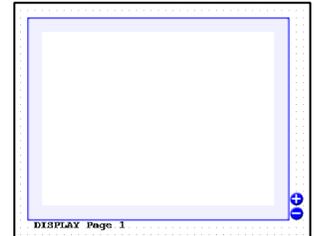


## 2.12 Parts Descriptions (Cont.)

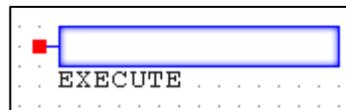
**CANTX**- CAN system transmitter



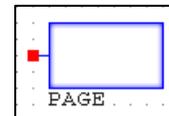
**Display**- Shows program output and simulates physical display (320x240)



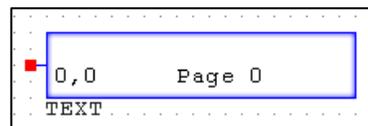
**Execute**- Can run an external file saved on SD card



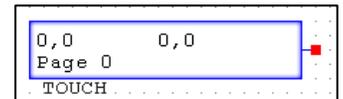
**Page**- Changes display pages based on numerical input  
Needs positive non-zero input to select draw page



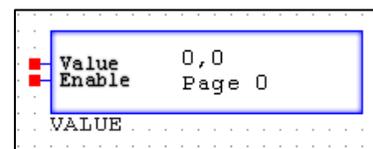
**Text**- Write text to the display at (X,Y) coordinates



**Touch**- Creates area on display that accepts touch input at (X,Y) coordinates

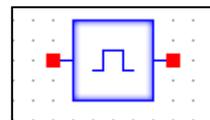


**Value**- Takes in numerical data and outputs it to the display at (X,Y) coordinates when enabled

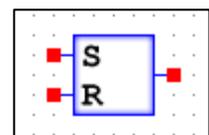


### Flops:

**Pulse**- Outputs a brief pulse when enabled



**Set Reset**- SR flip flop that can set or reset a Boolean output



## 2.12 Parts Descriptions (Cont.)

### Gates:

**AND(2,3,4,5)**- AND gate with varying inputs

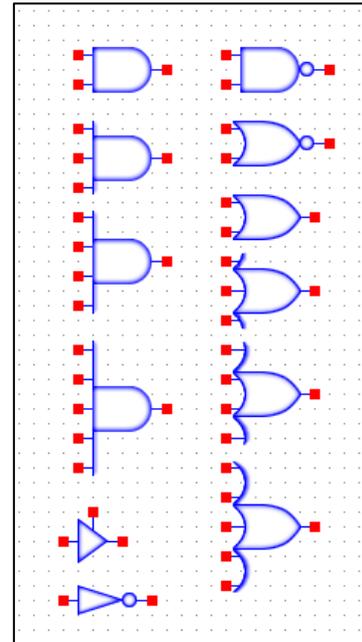
**Buffer**- Tri-State buffer with active high control

**Nand**- Functions as Nand gate

**Nor**- Functions as Nor gate

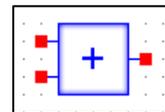
**Not**- Functions as Not

**OR(2,3,4,5)**- OR gate with varying inputs

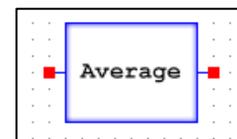


### Math:

**Add**- Adds two integer values (top is added to bottom)

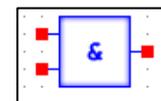


**Average**- Outputs average of inputting values based on power property ( $2^x$ ) so 0=average 1, 1=average 2, 6=average 64, where the resultant is the number of samples



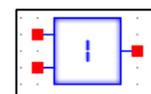
**BitAND**- AND operation of two integers in bit form. Used for bit masking to determine whether a bit is set or cleared.

EX: 0010 and 0010 becomes 0010



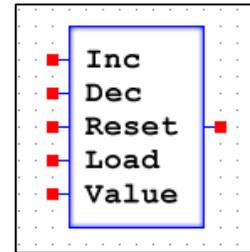
**BitOR**- OR operation of two integers in bit form. Used for bit masking to determine whether a bit is set or cleared.

EX: 0101 or 0011 becomes 0111

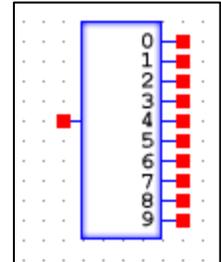


## 2.12 Parts Descriptions (Cont.)

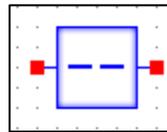
**Counter**- Increment/Decrement value with reset and load functionality. The range property can be used to prevent overflow and maintain a certain set of numbers



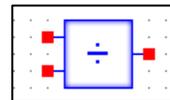
**Decode**- 1 to 10 decoder that selects based on integer input



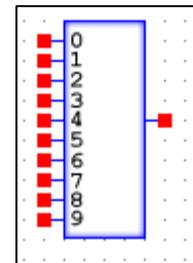
**Decrement**- Decreases integer value



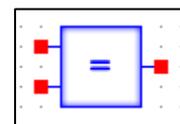
**Divide**- Divides two integer values (top is divided by the bottom)



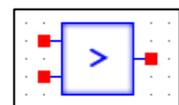
**Encode**- 10 to 1 encoder that outputs input that is true  
Outputs true as an integer value



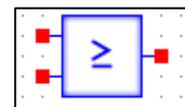
**Equal**- Checks two values to be equal and outputs Boolean value



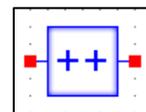
**Greater**- Check value to be greater than the other and outputs Boolean value (Top is greater than bottom input)



**GreaterEqual**- Checks value to be greater than or equal to the other and outputs Boolean value (Top is greater than or equal to bottom input)

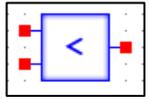


**Increment**- Increases integer value

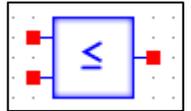


## 2.12 Parts Descriptions (Cont.)

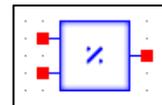
**Less**- Check value to be less than the other and outputs Boolean value  
(Top is less than bottom input)



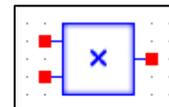
**LessEqual**- Checks value to be less than or equal to the other and outputs Boolean value (Top is less than or equal to bottom input)



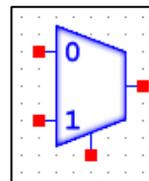
**Modulus**- Modulus division of two integers  
(Top is modulusly divided by bottom input)



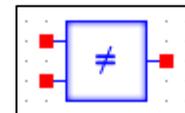
**Multiply**- Multiplies two integers  
(Top is multiplied by the bottom input)



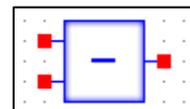
**Mux**- 2 to 1 Mux with active high control



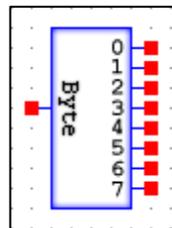
**NotEqual**- Checks values to be not equal and outputs Boolean value



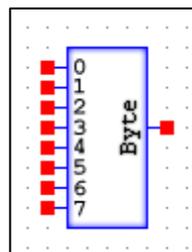
**Subtract**- Subtracts two integers  
(Top is subtracted by the bottom input)



**ToBits**- Unpackages byte into bits



**ToByte**- Packages bits into a byte



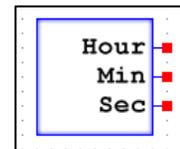
## 2.12 Parts Descriptions (Cont.)

### Timers

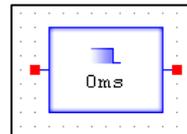
**Calendar**- Get month, day, and year from system clock  
(System clock used for PC simulation and RTC (real time clock) is used when executing on hardware)



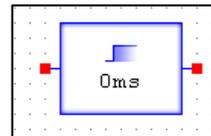
**Clock**- Get hours, minutes, and seconds from system clock  
(System clock used for PC simulation and RTC (real time clock) is used when executing on hardware)



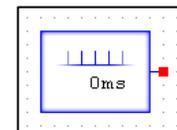
**DelayOff**- Delay on the falling edge



**DelayOn**- Delay on the rising edge

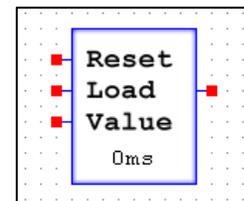


**Timer**- Outputs a pulse every specified time period  
Timer is 50% duty cycle, half on/half off



EX: 100ms period = 50ms on, 50ms off

**TimerCounter**- Counts up in set range according to set timer



# Chapter 3 – The Basics

## 3.1 Printing to Display

---

The basics to any form of programming utility starts with a “Hello World” problem. At the end of this Hello World tutorial, you will be to print to a display using a background and bitmap parts. Also, create touchable areas, make multiple pages, display text, and run the software simulator.

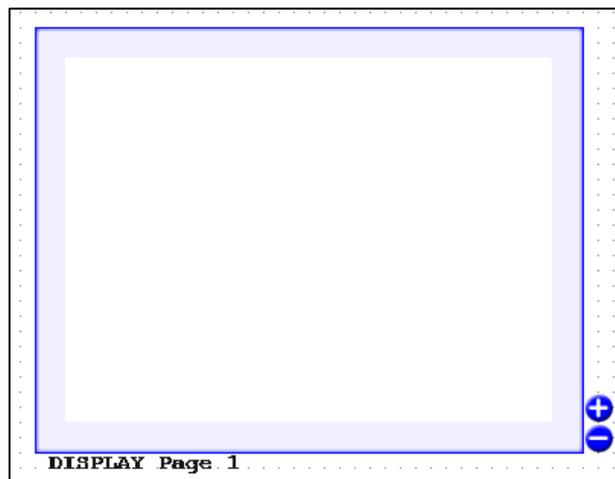
**Note:** Make sure the image library is set up before continuing.

1. Start up the CANcreate application and a new, blank project workspace will be displayed. If it is already open, make sure it's a blank project by clicking to **File->New**.

2. Go to **tools->Draw Part** and the parts window will pop up or press [P] on the keyboard

**Note:** Can also right-click anywhere on the workspace and select Draw Part.

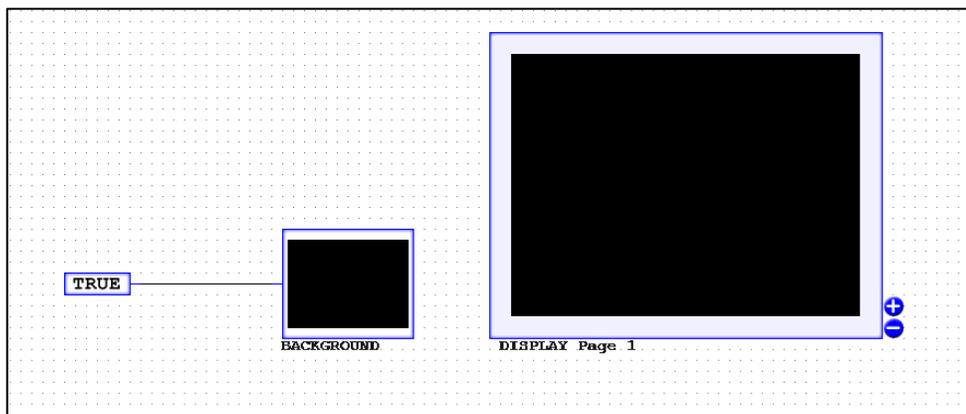
3. Under the drop down menu, go to Display. From there, select DISPLAY. There now should be a transparent blue rectangle following your pointer. Place the display into the workspace by left-clicking the grid field. The cursor can be used on the DISPLAY part to display the (X,Y) coordinate in the lower right corner.



4. Return to the Select Part window and under the Display section, select BACKGROUND. Place background into the workspace.

5. Exit the Select Part window and go to **tools->Select** to return to the pointer.
6. Double-click the background part and the BACKGROUND Properties window will appear. Change the Draw Page value to 1. (If set to 0, the Background would show up on all pages) Under Folder, select User/Manual and then “blackbg” under the image section. Click OK. Observe in the Display, a black background should now be shown instead of the red X.
7. Open up the Draw Part window and navigate to the Constants section. Select TRUE and place it in the workspace, near BACKGROUND. Switch to Draw Wire (**Tools->Draw Wire**). Click the red output node of TRUE and then click the red input node of BACKGROUND.

**Result:** Following steps 1 to 7 is seen below.

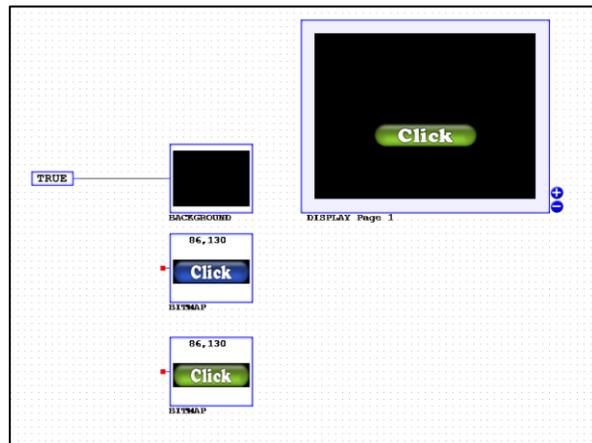


## 3.2 Using Bitmaps

---

8. Go to **Tools->Draw Part ->Display**, select BITMAP. Place a BITMAP part into the workspace. Change to select tool and double-click BITMAP. Set DrawX to 90 and DrawY to 125. Next, set Draw Page to 1 and locate User/Manual under the Folder drop-down menu. In image, find click\_btn. Click OK. The button image now is shown in the BITMAP part and on the Display.  
**Note:** The BITMAP location can also be edited by clicking and dragging the image on the Display. Arrow keys can also be used to fine adjust.
9. Repeat step 8, except this time under the image drop-down menu, select clicked\_btn. This button will appear over the previous BITMAP.

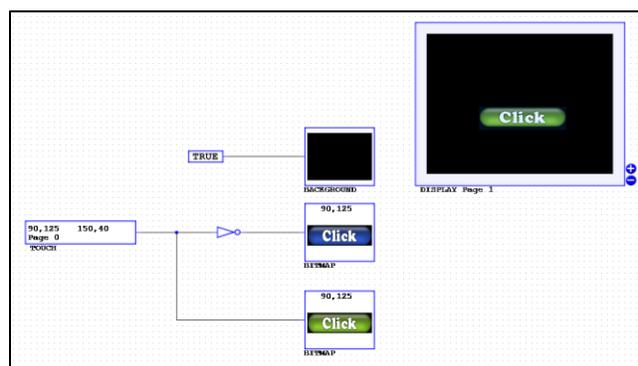
**Result:** Following steps 8 to 9 is seen below.



### 3.3 Using Touch

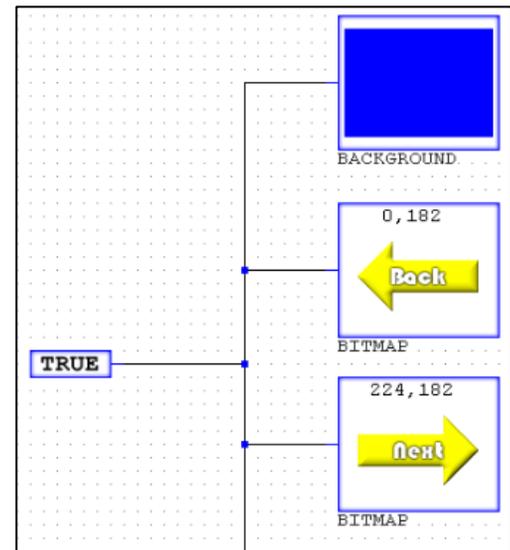
10. Next is to make that button click-able. Go to **Draw Parts->Display** and find TOUCH. Place a TOUCH part into the workspace. Switch to select and double-click the TOUCH part. Change DrawX to 90 and DrawY to 125. This is the location of the top left corner of the button. Change Width to 150 and Height to 40. This creates a touch area around the button that will recognize input. Finally change the Draw Page to 1. Click OK.
11. To have only one of the buttons displayed at a time, a NOT gate is needed. Go to **DrawParts->Gates** and select NOT from the list. Place it near the blue button. Switch to Draw Wires and connect the output of the NOT to the input of the blue button BITMAP. Now connect the TOUCH output to the NOT input and the green button BITMAP input.

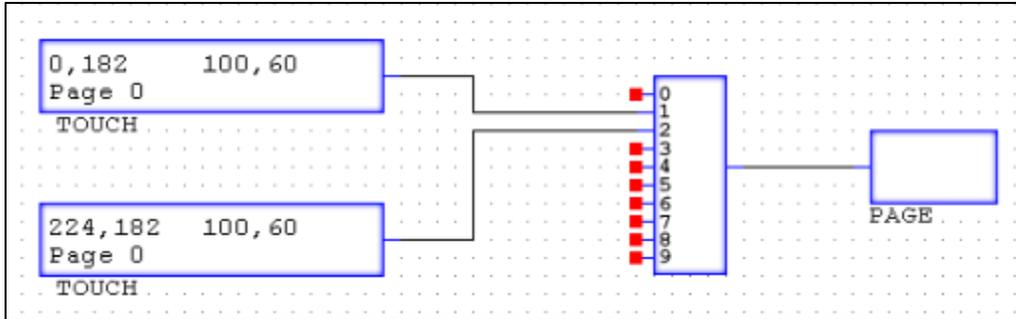
**Result:** Following steps 10 to 11 is seen below.



## 3.4 Creating Pages

12. To navigate between pages, more buttons are needed. Go to **Parts->Display** and select BITMAP. Under Part properties, set Draw X to 0, Draw Y to 182, and Folder to User\Manual. Next, set Image to Page\_Left. Make sure Page is set to 0 (0 so the BITMAP will be drawn on all pages) and transparent is checked. Click OK.
13. Insert another BITMAP. Under part properties, set Draw X to 224, Draw Y to 182, and Folder to User\Manual. Next, set Image to Page\_Right. Make sure Page is set to 0 and transparent is check. Click OK.
14. Connect a TRUE constant to both newly created BITMAPs.
15. These BITMAPs need touch areas. Go to **Parts->Display** and select TOUCH. Set Draw X to 0, Draw Y to 182, Width to 100, and Height to 60. Keep Draw Page 0, so that this touch area will appear on all the pages made. Click OK.
16. Insert a second TOUCH and go to the part properties. Set Draw X to 224, Draw Y to 182, Width to 100, and Height to 60. Keep Draw Page 0. Click OK.
17. To select between pages, an ENCODE part is used. Navigate to **Parts->Math** and select ENCODE. Place it near the two newly created TOUCH parts. Using the wire tool, connect the TOUCH at region (0,182) to node 1 on the ENCODE. Next, connect TOUCH at region (224,182) to node 2 on the ENCODE.
18. Finally, navigate to **Parts->Display** and select PAGE (Uses positive non-zero input to select the appropriate page to draw to the display). Insert PAGE near the output node of the ENCODE part. Change to wire and connect the ENCODE output node to the PAGE input

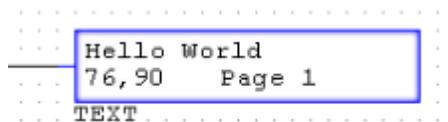




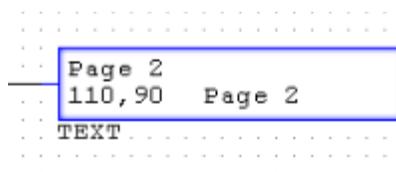
19. Put a background on page 2 by inserting **Parts->Display BACKGROUND**. Under part properties, change Draw Page to 2. Change folder to User\Manual and Image to “bluebg”. Click OK

### 3.5 Using Text

20. To add text to the display, navigate to **Parts->Display** and select TEXT. Insert it and go to part properties. Set Draw X to 76 and Draw Y to 90. Next, set Draw Page to 1. Under Text, type “Hello World”. In the 3 boxes above the Foreground button, set each to 255. In the 3 boxes above the Background button, set each to 0. Make sure transparent is unchecked. Click OK.



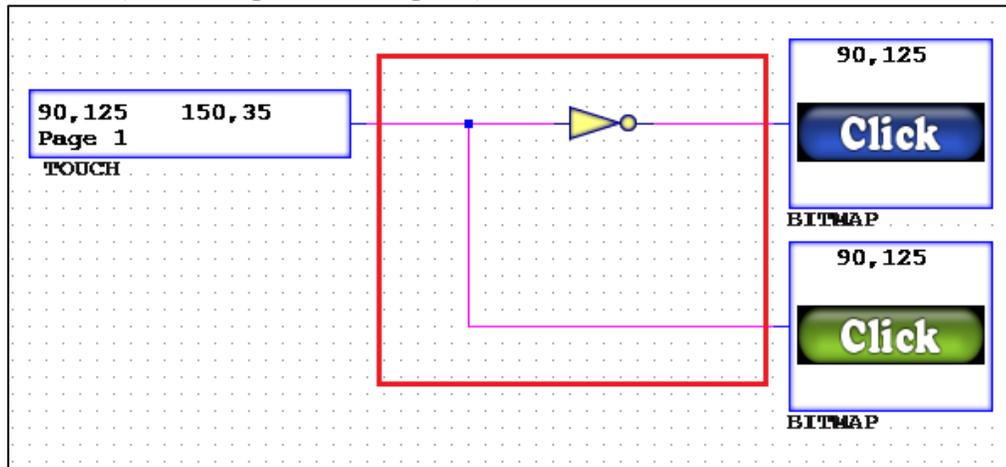
21. Insert another TEXT part and go to part properties. Set Draw X to 110, Draw Y to 90, and Draw Page to 2. In the text section, type “Page 2”. Above the Foreground button, set each box to 255. Above the Background button, set Red and Green to 0 and Blue to 255. Click OK.



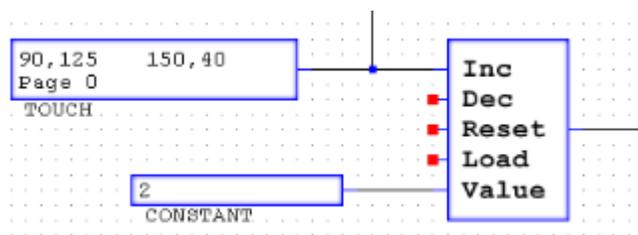
22. Connect a TRUE constant to each of those TEXT parts.

### 3.6 Using a Counter and Math Parts

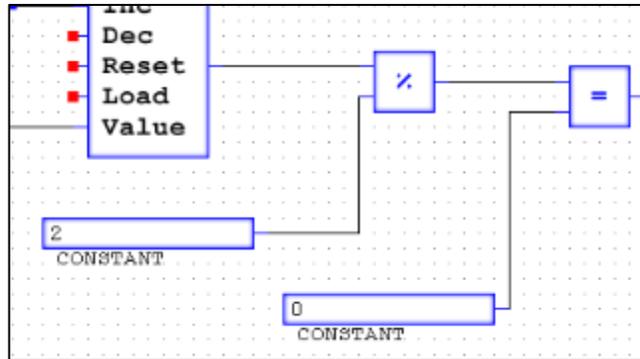
23. In this section, a counter and some simple gate combinations are going to be used to display and hide the “Hello World” text. First, draw a box and highlight the wires between the TOUCH (90,125) and the 2 Click-button BITMAPS (Including the NOT gate). Press Delete.



24. Attach a TRUE constant to the blue button BITMAP.
25. Now go to **Part->Math** and select COUNTER. Insert it next to the TOUCH (90,125). Change to wire and attach the TOUCH to the Inc node.
26. Go to **Part->Constants** and select CONSTANT. Insert it near the COUNTER and connect it to the Value node. Under part properties, change Constant to 2. Click OK.

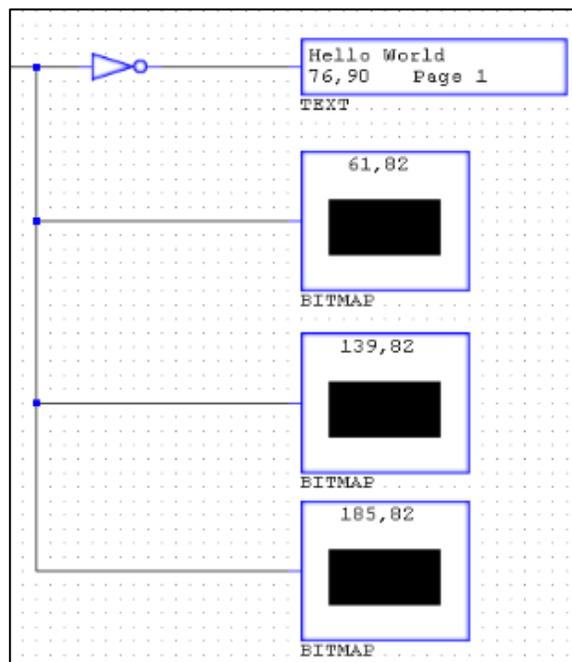


27. Return to **Part->Math** and select MODULUS. Connect it to the output of the COUNTER. For the second input node of MODULUS, insert a CONSTANT part and under part properties change Constant to 2. Click OK.
28. Next, in **Part->Math**, select EQUAL and insert it near the output of MODULUS. Change to wire and connect them. For the second input node of EQUAL, insert a CONSTANT part and under part properties change Constant to 0 (By default the value should be 0). Click OK.

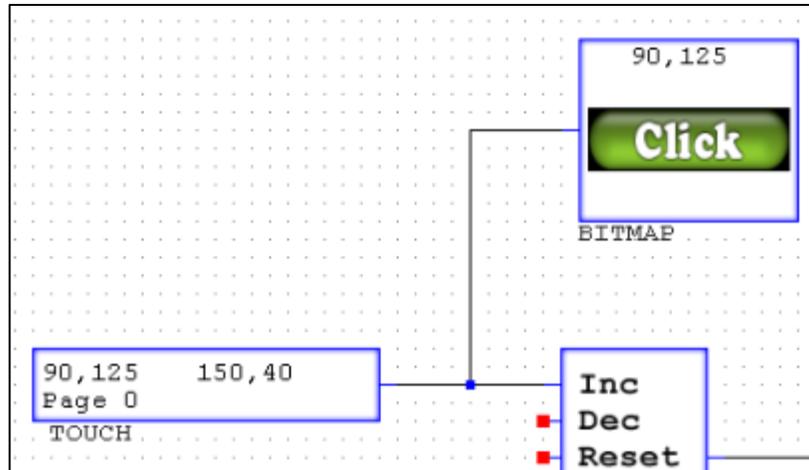


29. Go to **Part->Gates** and select NOT. Attach NOT to the output of EQUAL. Delete the TRUE connect to the "Hello World" TEXT part. Then attach the output of NOT to the "Hello World" TEXT.

30. Now insert 3 BITMAPS, first one is Draw X=61, Draw Y=82, Draw Page=1, Folder User\Manual, and Image "blank". Repeat for the next one, except change Draw X to 139 and Draw Y to 82. For the final one, repeat again except with Draw X=185 and Draw Y=82. Change to wire tool and connect each of these BITMAPS to the wire between EQUAL and NOT.



31. Finally, delete the TRUE connected to the green button BITMAP and connect the button BITMAP to the wire between TOUCH (86,130) and COUNTER.

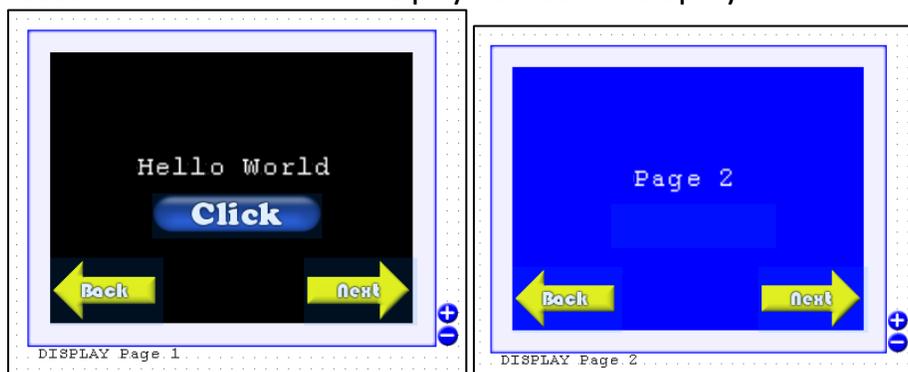


### 3.7 Running the simulator

32. Time to test the program. First, turn on the simulator. Either click the button in the quick access toolbar or press [F8].
33. Once the simulator is running, the wires will bold with either black, green, or blue colors.
34. Scrolling the pointer over the wire, the box in the lower right corner near the where the Page number is, will tell the value on the wire. For example, roll the pointer over the wire between the TRUE and the black BACKGROUND. This wire will be true and hence is green. Roll over any black wire and that will show as false.



35. In simulator mode, the DISPLAY screen in the workspace can be clicked and interacted with as if it were the physical touch display.

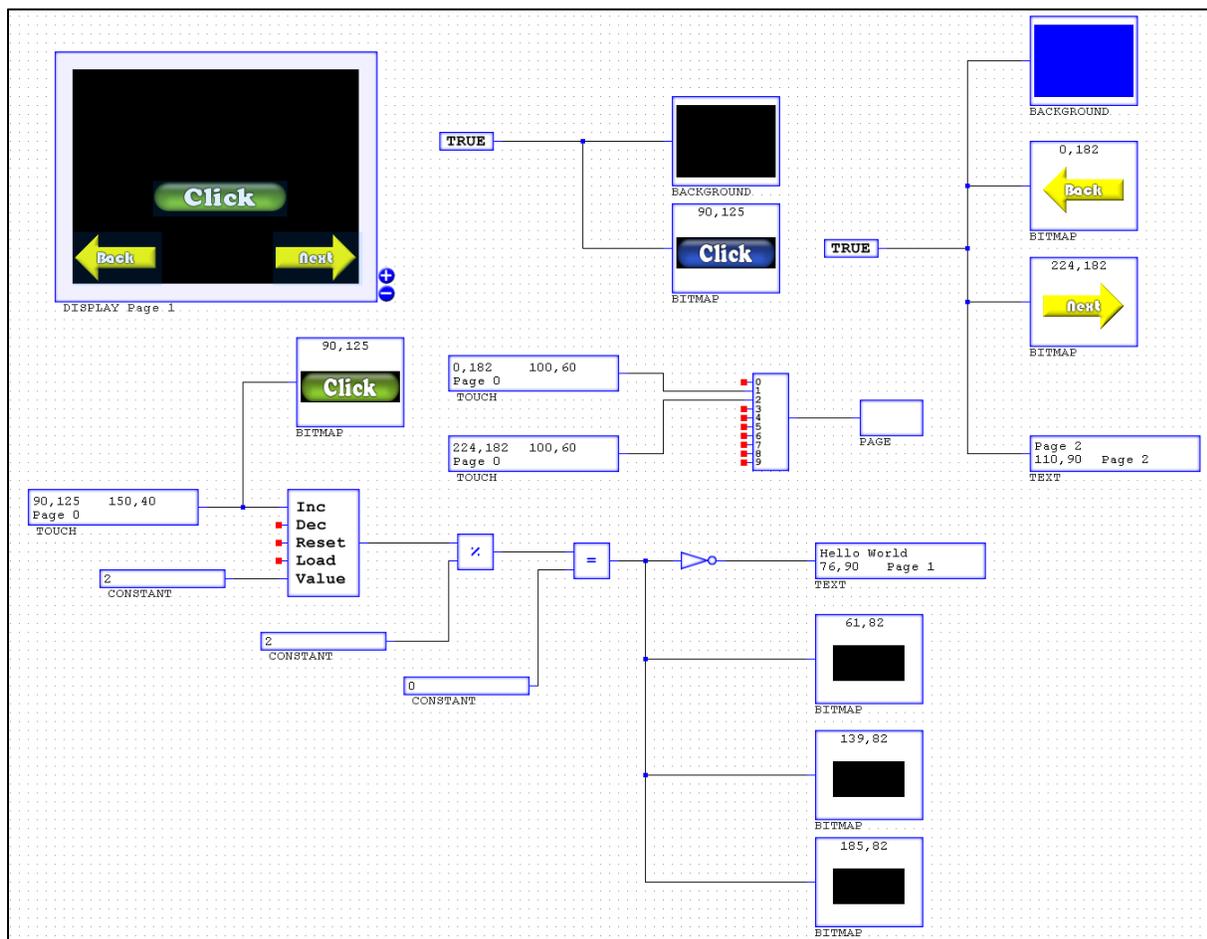


36. Clicking the blue button will result in the button confirming it was clicked

by flashing green and “Hello World” will be displayed on the screen. Notice the changing in values along the wires in the workspace. Click the blue button again to hide “Hello World”. Clicking next will go to Page 2 and clicking back will return to Page 1.

37. Click the simulator button or [F8] to turn off the simulation.

### 3.8 Result

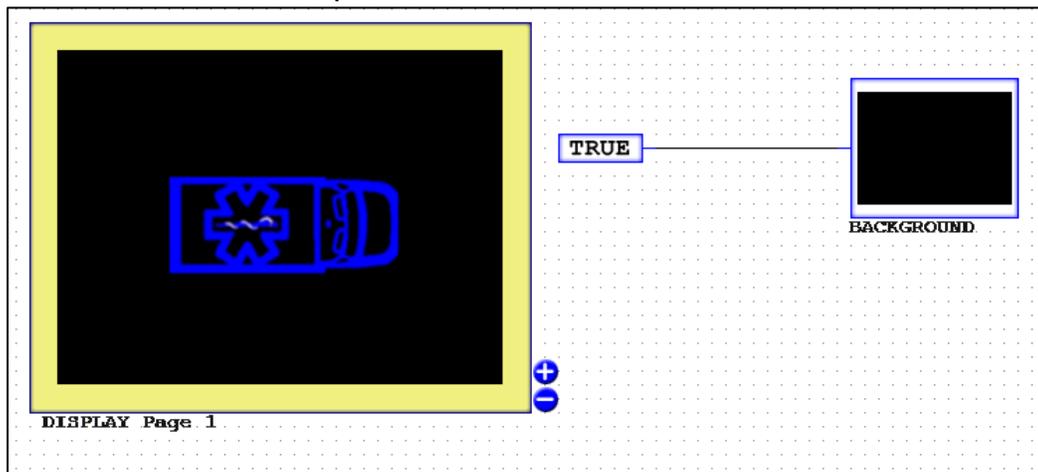


# Chapter 4 – Basic Animation

## 4.1 Blinking

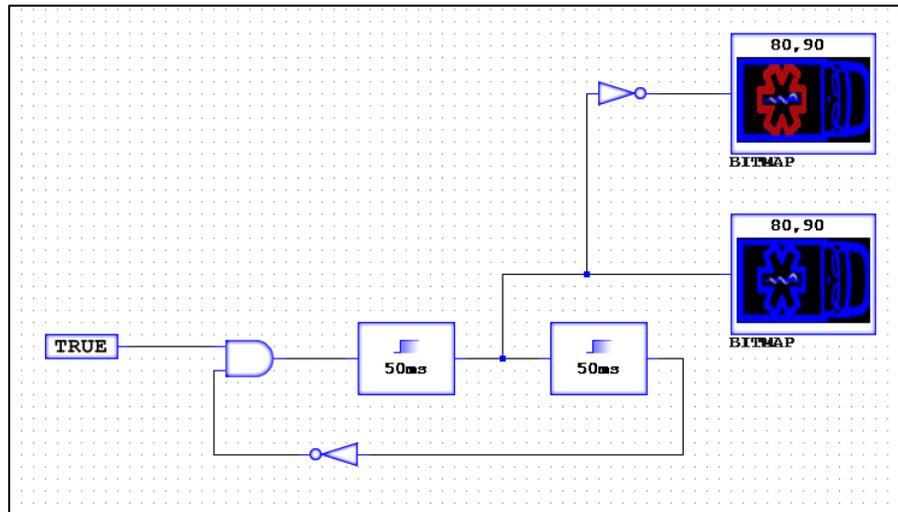
At the end of this Blinking tutorial, you will be able to create a basic bitmap animation that blinks on the screen at a certain rate.

1. Open up a new CANcreate workspace. Insert a DISPLAY.
2. Add in a BACKGROUND, under properties change Folder to User\Manual and Image to “blackbg”.
3. Insert a BITMAP and under properties set Draw X to 80 and Draw Y to 90. Next, change Folder to User\Manual and Image to “amb-red”. Connect a TRUE to BACKGROUND.
4. Now time to work on the blinking circuit. Go to **Part->Gates** and insert an AND2. To one of its inputs, connect a TRUE.

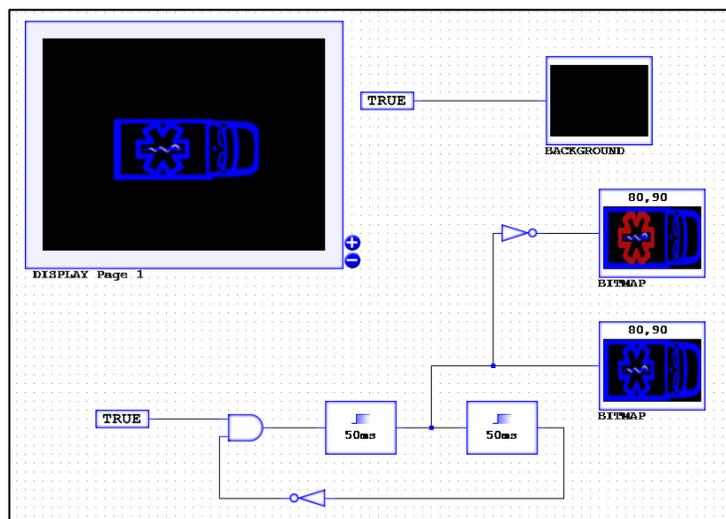


5. Next, go to **Part->Timers** and select DELAYON. Insert 2 of these into the workspace. Under properties for each DELAYON, change Timer to 50ms. Click OK.
6. Connect the output of the AND2 to the input of one of the 50ms DELAYONs. Now connect the output of that DELAYON to the input of the other DELAYON.

7. Go to **Part->Gates** and select a NOT. Insert the NOT, connecting its input to the last DELAYON's output. Finally, connect the output of the NOT to the second input of the AND2 gate.
8. Lastly, insert another BITMAP and under part properties set Draw X to 80 and Draw Y to 90. Next, change Folder to User\Manual and Image to "amb-blue". Click OK. Connect its input to the wire between the two DELAYONs. Insert another NOT gate and connect it to the wire going to the "amb-blue" and its output to the "amb-red".



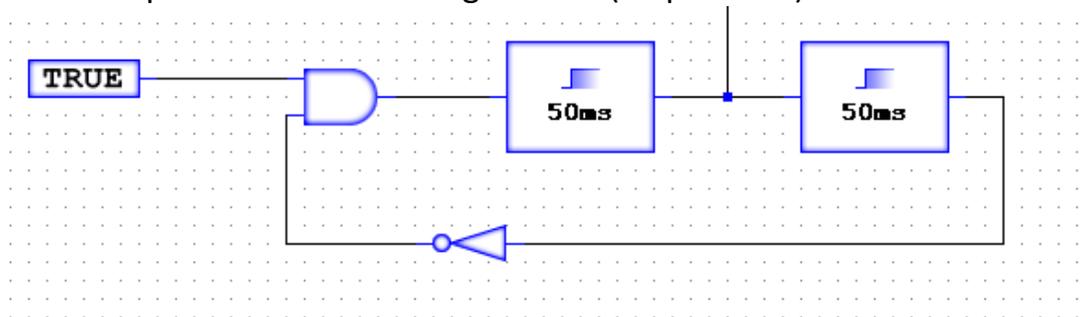
9. Run the simulation and watch the ambulance light blink.  
**Note:** The combination of these DELAYONs creates a 100ms delay, which is 10 frames per second. This is the **highest** recommended frame rate for the display.



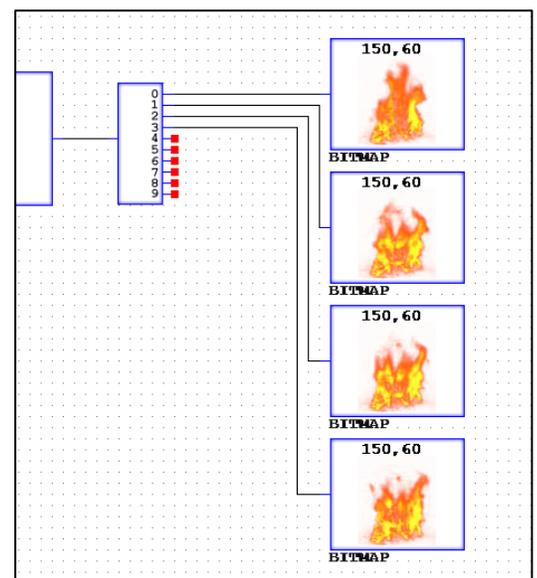
## 4.2 Image Slideshow

At the end of this Slideshow tutorial, you will be able to cycle through images to create a slideshow effect and simulate animation.

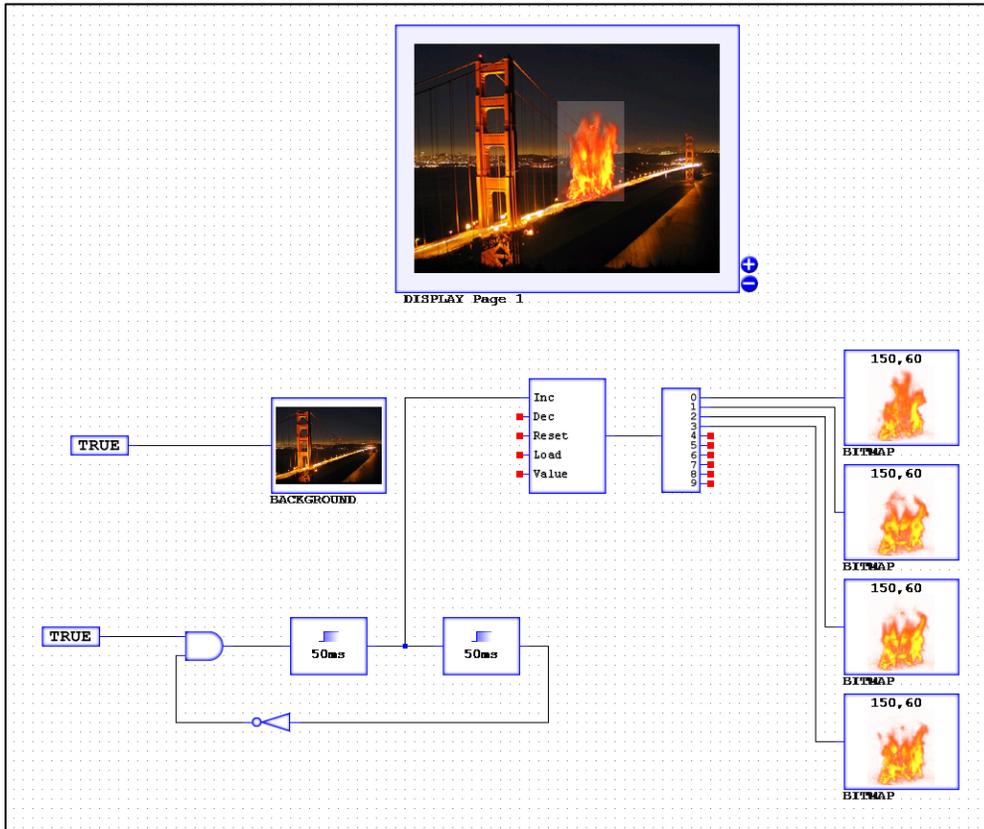
1. Open up a new CANcreate workspace and assemble the AND/DELAYON circuit explained in the blinking tutorial (steps 5 to 7).



2. Insert a DISPLAY and a BACKGROUND. Set the background to “bridge”.
3. Next, go to **Part->Math** and insert a COUNTER. Attach the Inc node on the COUNTER to the wire between the two DELAYONS.
4. Now go to **Part->Math** and select a DECODE. Insert it next to the COUNTER. Under COUNTER properties, set range to 3. Attach the output of the COUNTER to the input of the DECODE.
5. Insert 4 BITMAPs, all at Draw X= 150 and Draw Y=60. For each, change Folder to User\Manual. Make sure transparent is checked. Set each BITMAPs' Image to “fire1”, “fire2”, “fire3”, and “fire4” respectively.
6. Connect “fire1” BITMAP to the node 0 output of the DECODE. Connect “fire2” to node 1, “fire3” to node 2, and “fire4” to node 3.



7. Run the Simulator and the fire animation should repeat.

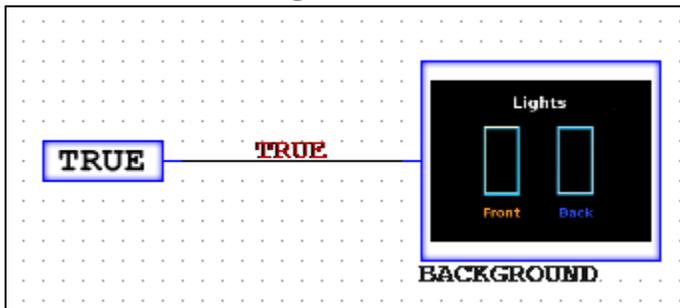


# Chapter 5 – Labels and Inputs

## 5.1 Labeling

At the end of this Labels and Inputs tutorial, you will be educated on using wire labeling correctly and efficiently. You will also be able to create multiple function touch inputs.

1. Open up a new CANcreate workspace and insert a DISPLAY.
2. Now insert a BACKGROUND and under User/Manual, set Image to “Lights”. Wire a TRUE to the BACKGROUND.
3. Next, switch to Place Label and click the wire between TRUE and BACKGROUND. In the drop-down menu, type “TRUE”. Click OK. This same label can now be applied to other wires, elsewhere, and then will take on the state of this original labeled wire.

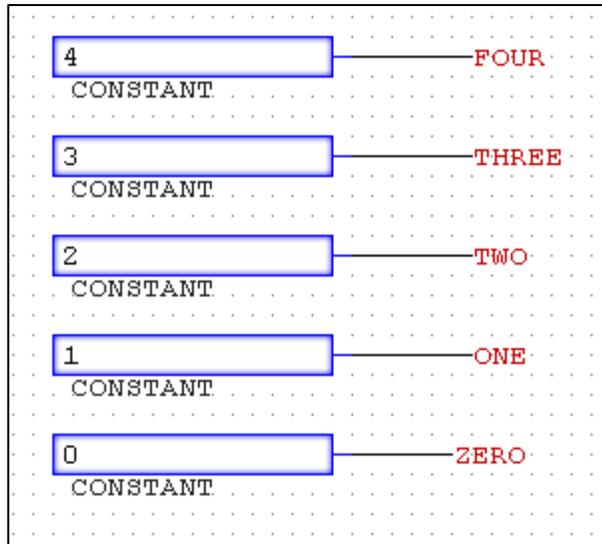


**Note:** Can also double-click on wire while in Select tool.

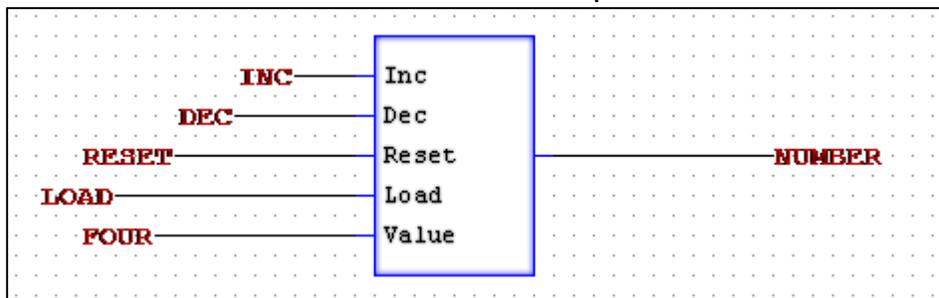
4. Insert 5 BITMAPS, with the following X and Y coordinates. Each will have the same image from User\Manual and Image “front\_line”.
  1. Draw X=80, Draw Y=140
  2. Draw X=80, Draw Y=125
  3. Draw X=80, Draw Y=110
  4. Draw X=80, Draw Y=95
  5. Draw X=80, Draw Y=80
5. Draw a wire from each BITMAP without attaching any of them to another part. Double-click the wire from (80,140)BITMAP and under the drop down menu, type “F\_0”. For the (80,125)BITMAP, double-click the wire and label it “F\_1”. For the (80,110)BITMAP, label the wire “F\_2”. For the

(80,95)BITMAP, label the wire "F\_3". Finally, for the (80,80)BITMAP, label the wire "F\_4".

- Before creating the counter, constants need to be established. Insert 5 CONSTANT parts. Set each respectively, as values 0, 1, 2, 3, and 4. Next, attach open ended wires to each and label each wire corresponding to the value. So "ONE" for the 1 value, "TWO" for the 2 value, and so on.

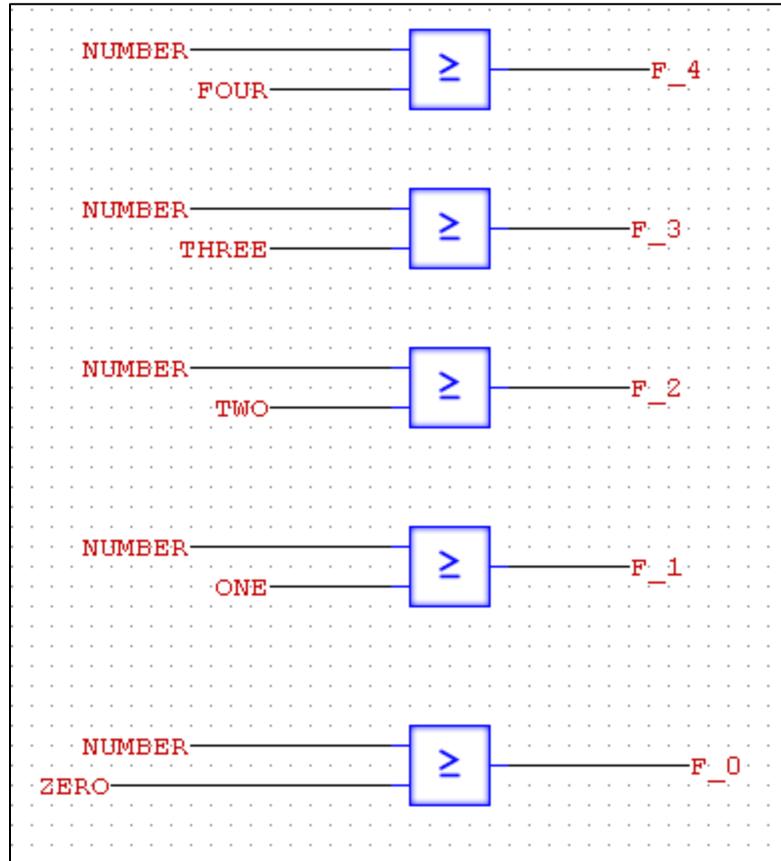


- Under Part->Math, select a COUNTER and insert it in the workspace. Attach open ended wires to all 5 inputs of the COUNTER. Label each wire with the following names: "INC" for Inc, "DEC" for Dec, "RESET" for Reset, and "LOAD" for Load. For the value wire, double-click the wire, and click the drop down menu to see the list of labels. Select "FOUR" to use that label and associate this wire with the CONSTANT part FOUR.



- Draw a wire from the output of the COUNTER. Label it "NUMBER". Now go to **Part->Math** and select GREATEREQUAL. Insert 5 of these. Attach open ended wires to both the inputs and outputs of the GREATEREQUALs. Select 1 of them and using pre-existing labels; label the topmost input wire

“NUMBER” and the bottom one “FOUR”. The output wire should be labeled “F\_4”. Select the next GREATEREQUAL, label the inputs “NUMBER” and “THREE” and the output “F\_3”. Repeat for the next 3 GREATEREQUALs, for TWO, ONE, and ZERO using F\_2, F\_1, and F\_0.

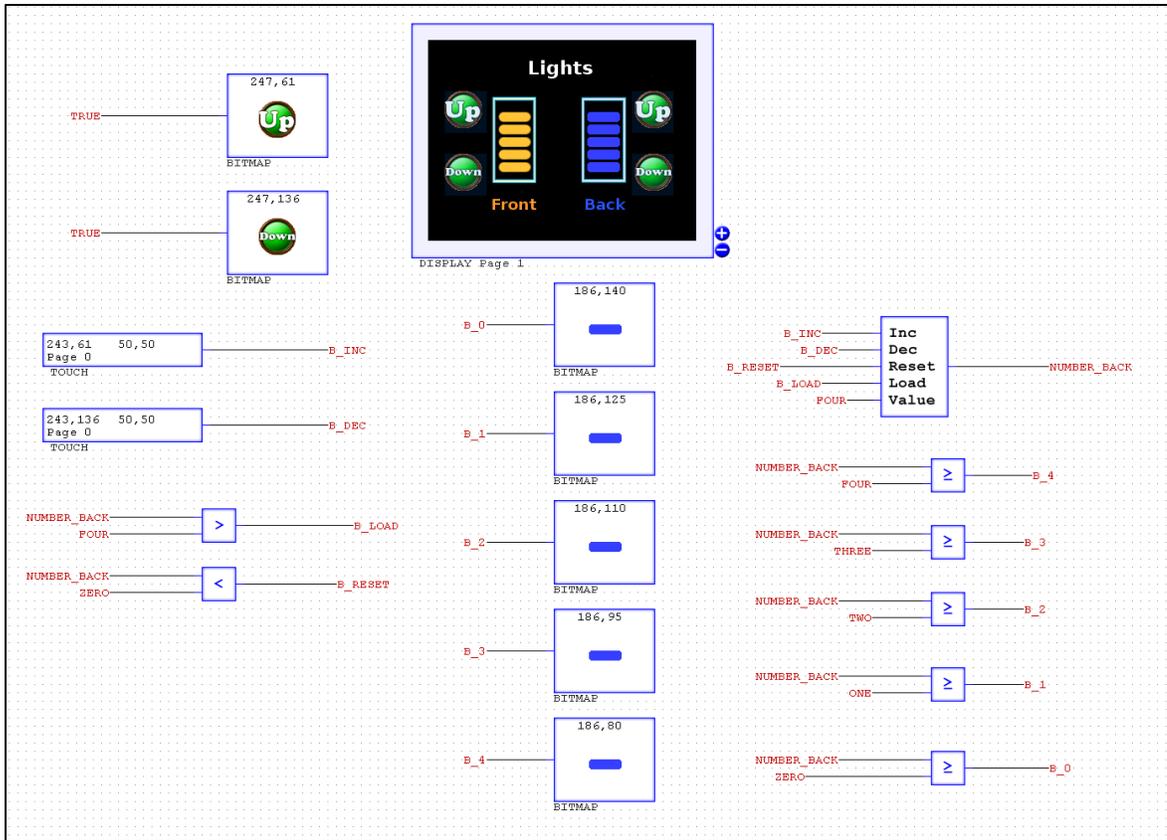


9. Next, is to create buttons to make the light level increase and decrease. Insert 2 BITMAPS, set them both to TRUE. For one, set Draw X to 20 and Draw Y to 61. Under Folder, go to User\Manual and set Image as “up\_button”. For the second BITMAP, set Draw X to 20 and Draw Y to 136. Under Folder, go to User\Manual and set Image as “down\_button”.
10. Insert a TOUCH part. Set Draw X to 20, Draw Y to 61, Width to 50 and Height to 50. Click OK. Attach a wire to its output and label it “INC”. Insert a second TOUCH part, set Draw X to 20, Draw Y to 136, Width to 50, and Height to 50. Click OK. Attach a wire to its output and label it “DEC”.
11. Run the simulator, press Up to increase the bars and down to decrease the bars.

## 5.2 Extension: More Labeling

---

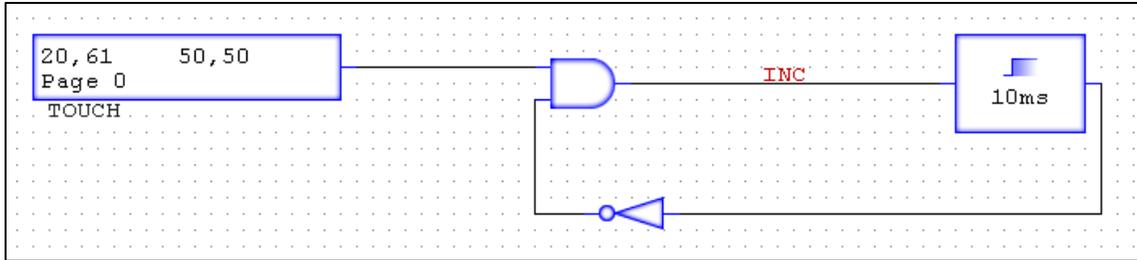
12. Now repeat steps 4 to 11 with the back light controls (Skip step 6). Refer to the following steps to get new names and locations.
13. For the 5 BITMAPS use Image “back\_line” and these locations:
  1. Draw X=186 Draw Y=140
  2. Draw X=186 Draw Y=125
  3. Draw X=186 Draw Y=110
  4. Draw X=186 Draw Y=95
  5. Draw X=186 Draw Y=80
14. Label them using labels “B\_0”, “B\_1”, “B\_2”, “B\_3”, and “B\_4”.
15. For the counter, use labels “B\_INC” for Inc, “B\_Dec” for Dec, “B\_Reset” for Reset, “B\_Load” for Load, and continue to use “FOUR” for Value. The output label should be “NUMBER\_BACK”.
16. Recreate the GREATEREQUAL functions except replace “NUMBER” with “NUMBER\_BACK”, and the output from “F\_0” to “B\_0” and so on.
17. For the LESS and GREATER parts, just replace “NUMBER” with “NUMBER\_BACK”. Make the GREATER output “B\_Load” and the LESS output “B\_Reset”.
18. Set the up button to Draw X=247 and Draw Y=61. Next, set the down button to Draw X=247 and Draw Y=136.
19. Finally, for the two TOUCH parts, set one to Draw X=243, Draw Y=61, Width=50, and Height=50. Then set the output to “B\_Inc”. For the second TOUCH part, set Draw X=243, Draw Y=136, Width=50, and Height=50.
20. Test in simulator to make sure this functions the same as the front light control.



### 5.3 Multi-Function Input (Short/Long Presses)

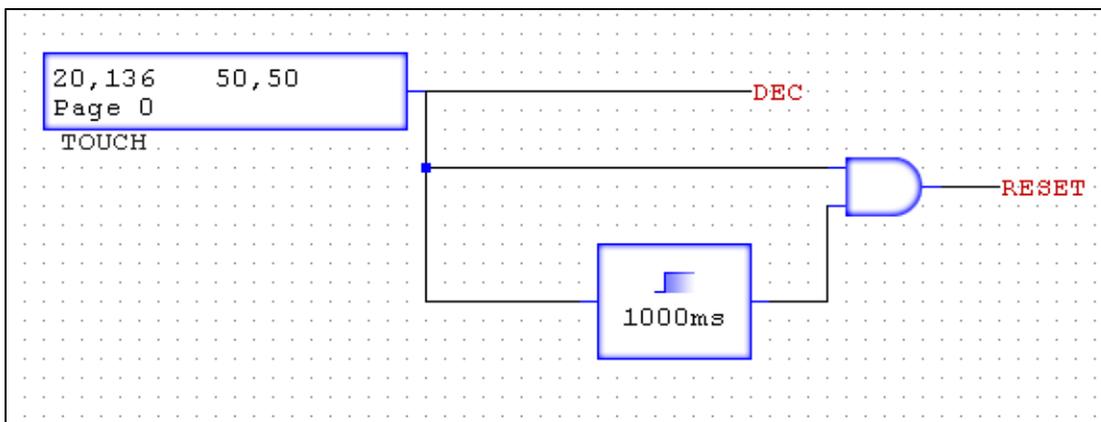
22. This will create multiple control options through the same button. For the front up button, holding the button will slowly increase the light and the down button, when held, will reset the light to zero.

23. Locate the TOUCH(20,61) that is outputting to "INC". Delete the output wire from the TOUCH. Now insert an AND2 gate, a DELAYON, and a NOT gate. Attach the output of the TOUCH to the AND2 input and the output of the NOT gate to the AND2 input. Draw a wire from the output of AND2 to the DELAYON. Next, wire the output of DELAYON to the NOT gate. Label the wire between the AND2 and the DELAYON as "INC". Finally, set the DELAYON timer to 10ms.



24. Run the simulator and hold the up button for the front; the bars should increase without clicking.

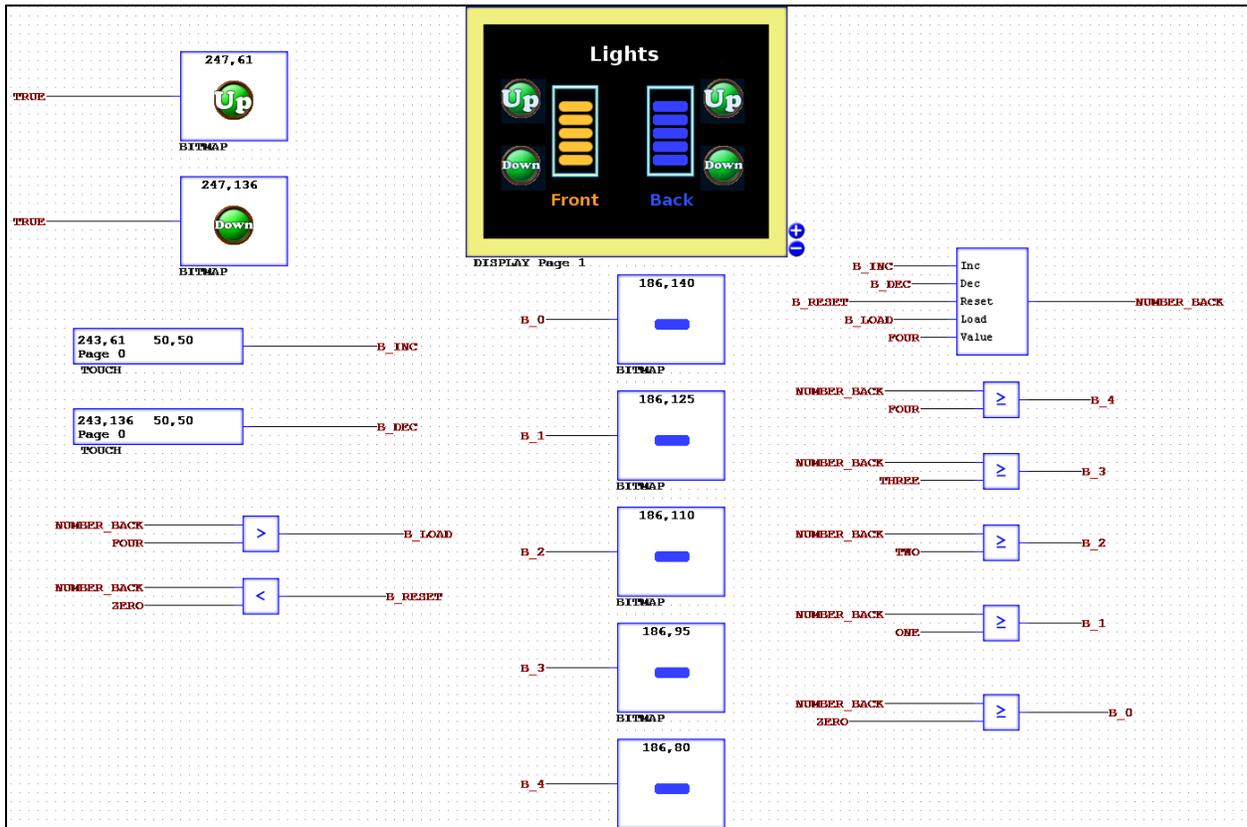
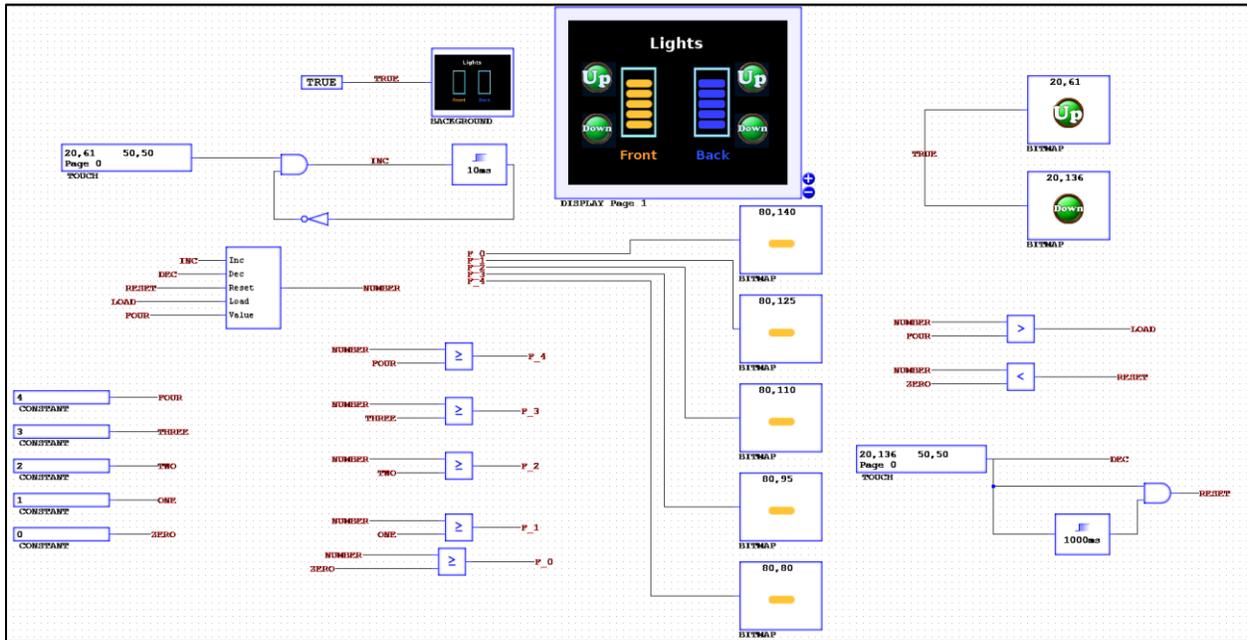
25. Next is to change the down button. Draw a wire from TOUCH(20,136) wire labeled "DEC". Insert an AND2 gate and a DELAYON. Attach one input of the AND2 to the newly created wire and also connect this wire to the input of the DELAYON. Connect the output of the DELAYON to the second AND2 input. Label the output of the AND2 as "Reset".



26. Run the simulator and test the long-press down button. No matter value it is at, when held for 1 second, it will reset to 1 bar.

**Note:** Although this program works correctly in the software simulation, if setup on hardware, it wouldn't. The light "bars" would not be erased from the screen because on hardware, BITMAPs are only erased by being overwritten. Blank black images would need to be drawn to the screen when decreasing the lights to get rid of the "bars" from the screen.

# 5.4 Result



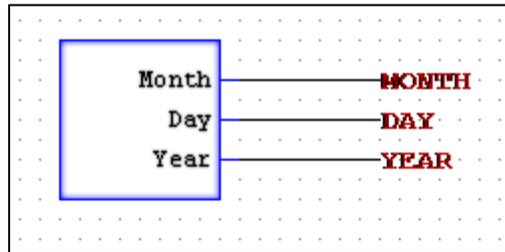
# Chapter 6 – calendar and Time

## 6.1 Calendar Part

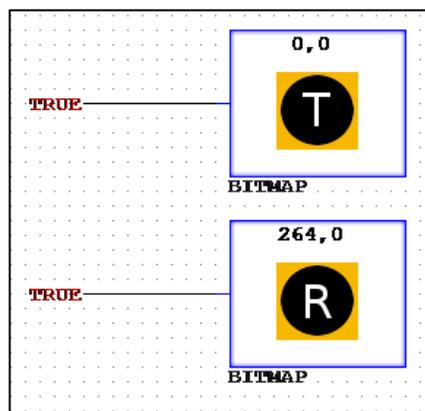
---

At the end of this Time tutorial, you will be able to better understand how to use the clock and calendar parts within both the simulator and on the physical device.

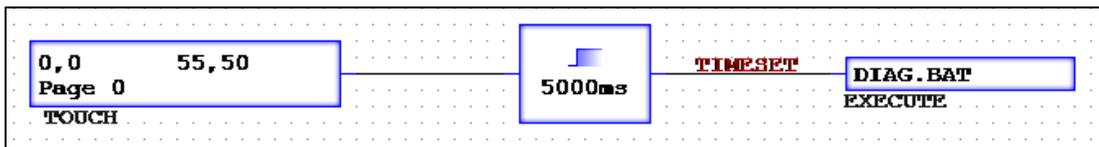
1. Open up a new CANcreate workspace and insert a DISPLAY.
2. Now insert a BACKGROUND and under User/Manual, set Image to “tire\_calendar”. Wire a TRUE to the BACKGROUND. Label that wire TRUE to use later.
3. Go to **Part->Timers**, and select CALENDAR. Attach wires to each of the outputs and label them (top to bottom) Month, Day, and Year.



4. Next, insert 2 BITMAPs. Set each to TRUE. For the the first BITMAP, Set Draw X to 0, Draw Y to 0, and change Folder to User\Manual. Set the Image to “tire\_time\_btn\_1”.Click OK. For the second BITMAP, set Draw X to 264, Draw Y to 0, and set Folder to User\Manual. Set the Image to “tire\_replace\_btn\_1”.Click OK.



- After that, insert 2 more BITMAPS. Set one to Draw X=0, Draw Y=0, and under User\Manual, Image to “tire\_time\_btn\_2”. Click OK. Set the second to Draw X=264, Draw Y=0, and under User\Manual, Image to “tire\_replace\_btn\_2”. Click OK.
- Insert a TOUCH at Draw X=0, Draw Y=0, Width=55, and Height=50. Attach a DELAYON to the output and set the timer to 5000ms (5 seconds). Now go to **Part->Display** and select EXECUTE. Attach this to the output of the DELAYON. Change the EXECUTE Filename to “DIAG.BAT”.



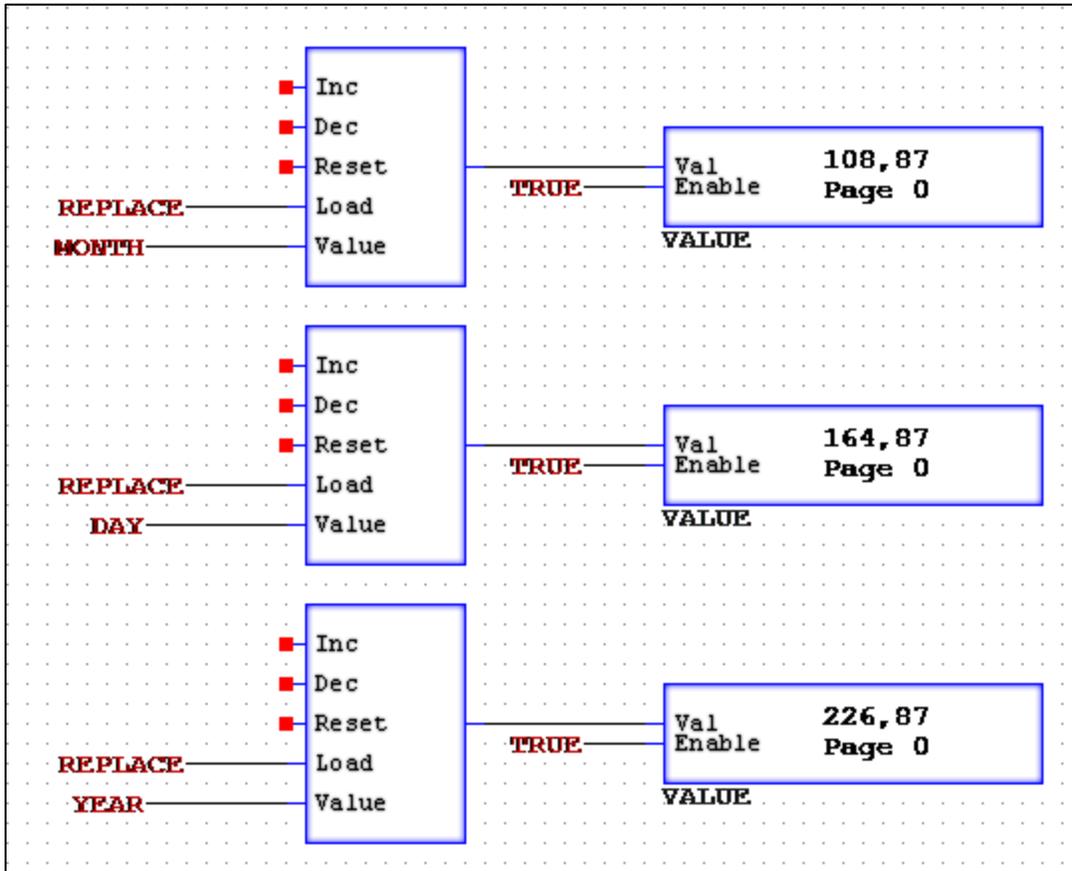
**Note:** This execution will only work once on the device, not in the simulator.

- Label the wire between the DELAYON and the EXECUTE as “Timeset”. Create a wire attached to the BITMAP(0,0) with the red T button and label it “Timeset”.
- Insert another TOUCH at Draw X=264, Draw Y=0, Width=55, and Height=50. Insert a DELAYON and attach it to the output. Then attach the output of the DELAYON to the BITMAP(264,0) with the red R button. Change the timer on the DELAYON to 5000ms. Finally, label the wire between the DELAYON and BITMAP as “Replace”.



- Now, insert 3 COUNTERS and 3 VALUE parts (under **Part->Display**). Set a labeled wire as “Replace” to each of the LOAD inputs. Next, set a labeled wire as “Month” to one of the COUNTER's Value input. Repeat with the other 2 COUNTERs using “Day” and “Year”. Wire a VALUE part to each of the outputs of the COUNTERs. Set each VALUE Enable to a TRUE connection.
- For the COUNTER with “Month” as Value, set the VALUE part to Draw X=108 and Draw Y=87. Then set Width=30 and Height=0. Set foreground to

(0,0,0) which is (red,green,blue). Set background to (246,182,0).Click OK. For the COUNTER with “Day” Value, set the VALUE part to Draw X=164 and Draw Y=87. Then set Width=30 and Height=0. Change foreground and background color to that previously stated. Click OK. For the final COUNTER with “Year” Value, set the VALUE part to Draw X=226 and Draw Y=87. Then set Width=50 and Height=0. Change foreground and background color to that previously stated. Click OK.



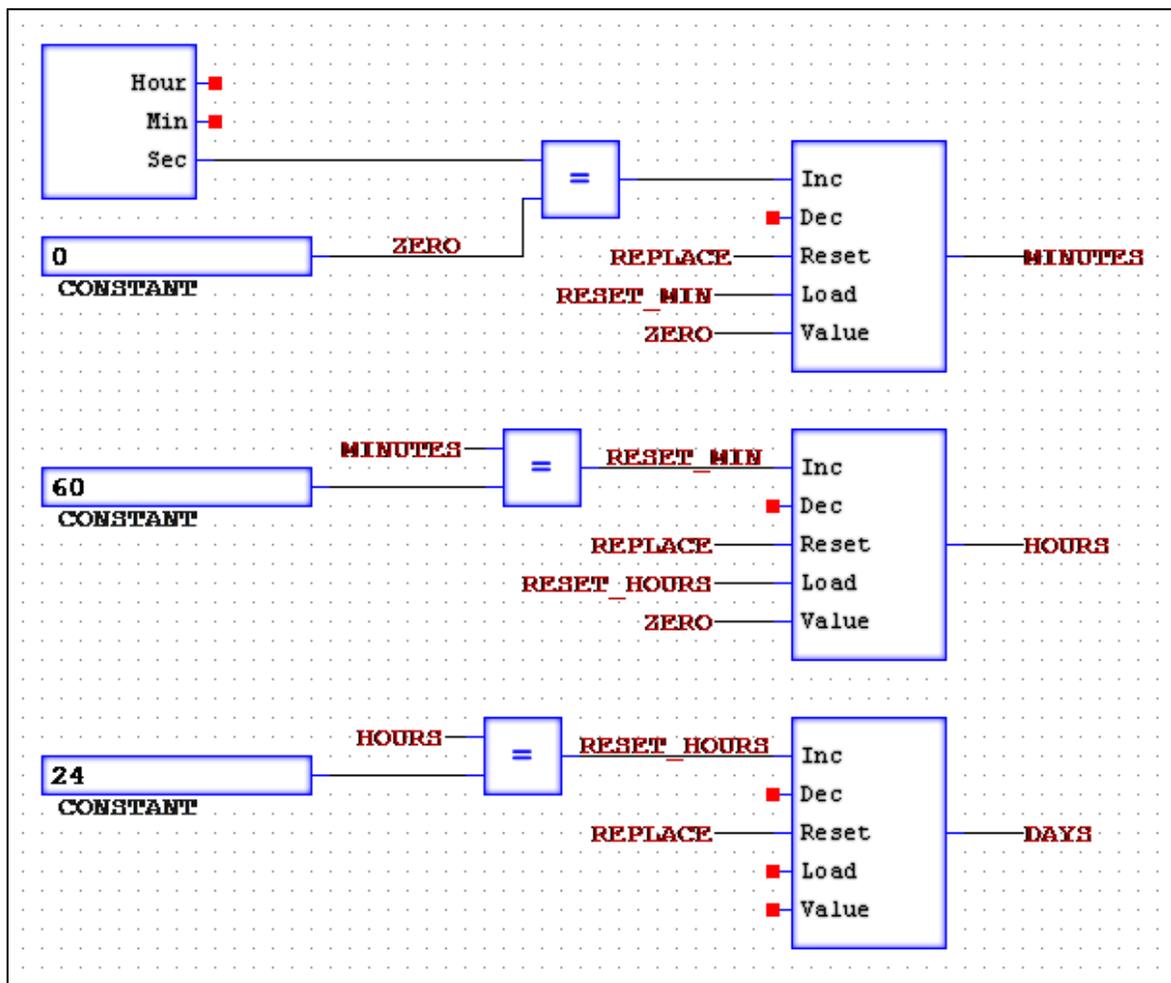
## 6.2 Clock Part

11. Under **Part->Timers**, select CLOCK. Next, insert 3 CONSTANTS, 3 EQUALS, and 3 COUNTERS. Set one CONSTANT to 0, second to 60, and the third to 24.
12. Connect the Sec output from CLOCK and the 0 CONSTANT to the EQUAL inputs. Label the wire from the 0 CONSTANT as “Zero”. Then connect that output to one of the COUNTERs. On that same COUNTER, set a wire

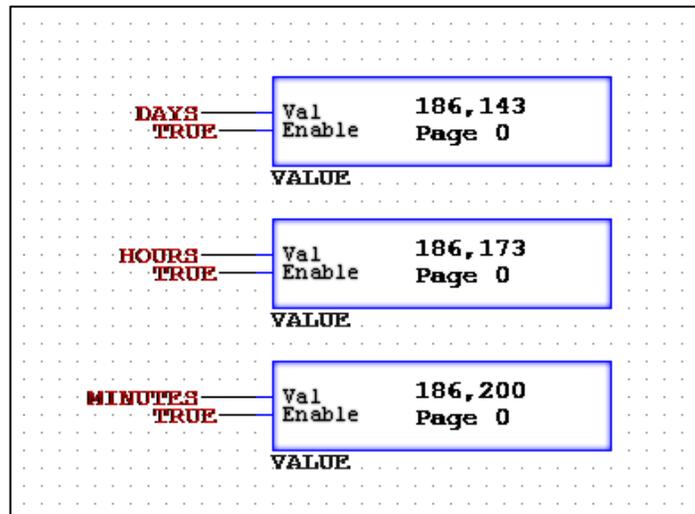
“Replace” to the Reset input. Set a wire “Reset\_Min” to the Load input and set a “Zero” wire to the Value input. Finally, label the COUNTER output wire to “Minutes”.

13. Using the second EQUAL and COUNTER, set a wire “Minutes” to one EQUAL input and the 60 CONSTANT to the second input. Connect the EQUAL output to the COUNTER Inc input. Set “Replace” wire to Reset, “Reset\_Hours” wire to Load, and “Zero” to Value. Finally, label the COUNTER output to “Hours”.

14. Using the last EQUAL and COUNTER, set a wire “Hours” to one EQUAL input and the 24 CONSTANT to the second input. Connect the EQUAL output to the COUNTER Inc input. Set “Replace” wire to Reset and label the COUNTER output as “Days”.

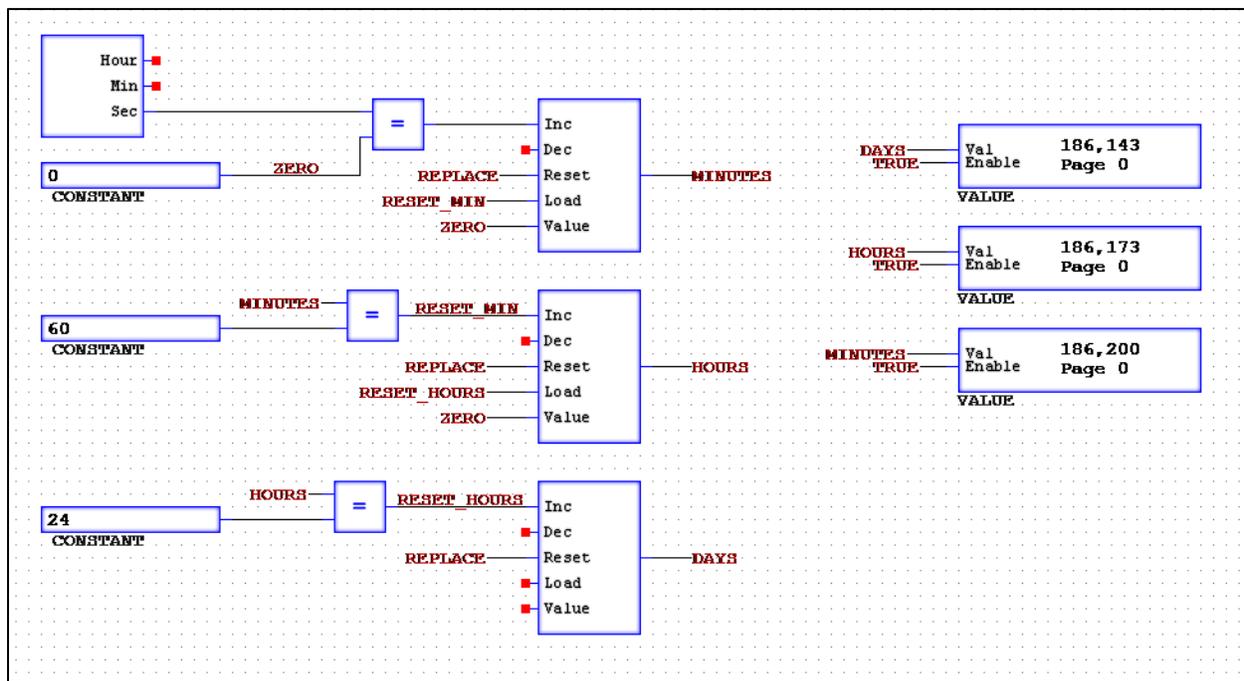
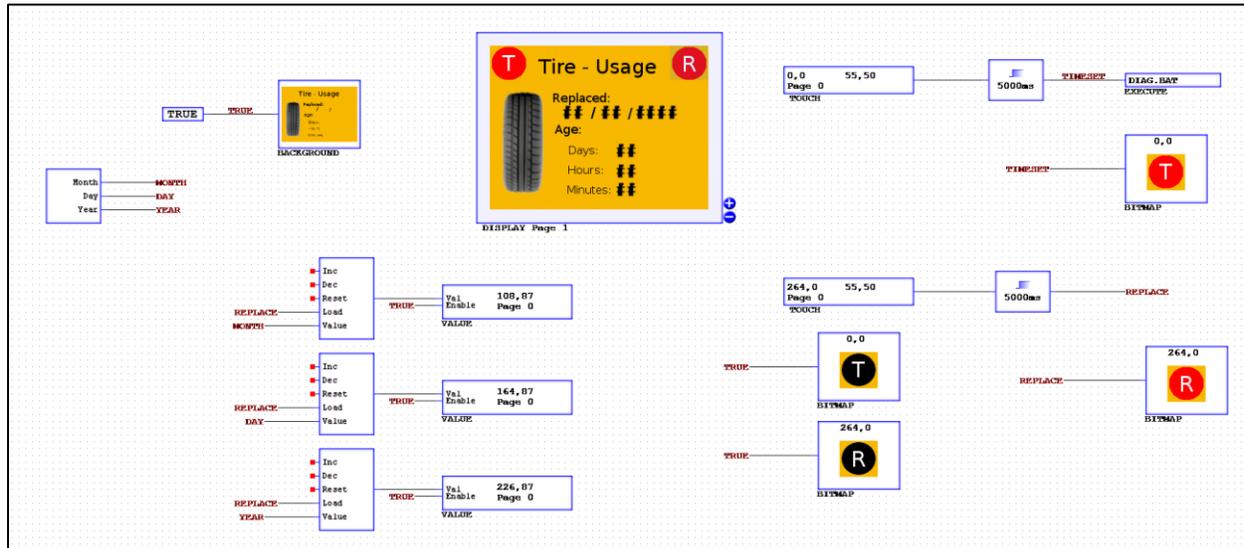


15. Finally, insert 3 VALUE parts into the workspace. Set each Enable to TRUE. For the first VALUE, set Draw X=186, Draw Y=143, Width=30, and Height=0. Set foreground to (0,0,0) which is (red,green,blue). Set background to (246,182,0). Click OK. Label the Value input to “Days” wire.
16. For the second VALUE, set Draw X=186, Draw Y=173, Width=30, and Height=0. Set the foreground and background color to that stated in step 15. Click OK. Label the Value input to “Hours” wire.
17. For the last VALUE, set Draw X=186, Draw Y=200, Width=30, and Height=0. Set the foreground and background color to that stated in step 15. Click OK. Label the Value input to “Minutes” wire.



18. Test with the simulator. Hold the R button for 5 seconds to “replace” the tire. The date replaced will appear and as each minute passes, the age will update. Holding the T button while on the device will open up the set time screen since the clock needs to be set on the device.

## 6.3 Result

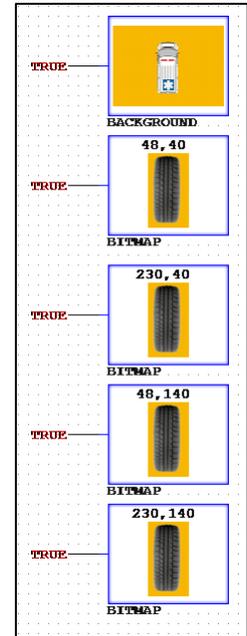


# Chapter 7 – Advanced Page Navigation

## 7.1 Multiple Page Setup

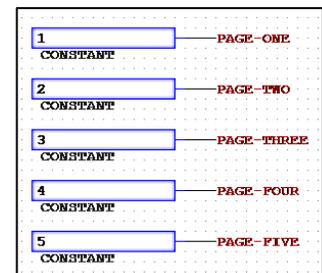
At the end of this Page Navigation tutorial, you will be able to handle multiple page programs better and create wrap around page navigation using buffers.

1. Open up a new CANcreate workspace and insert a DISPLAY.
2. Now insert a BACKGROUND and under User/Manual, set Image to “tires”. Wire a TRUE to the BACKGROUND.
3. Add 4 BITMAPS into the workspace. Each will use Folder User\Manual and Image “tireico1”. Use the following Draw X and Draw Y for each BITMAP. Set each to TRUE.



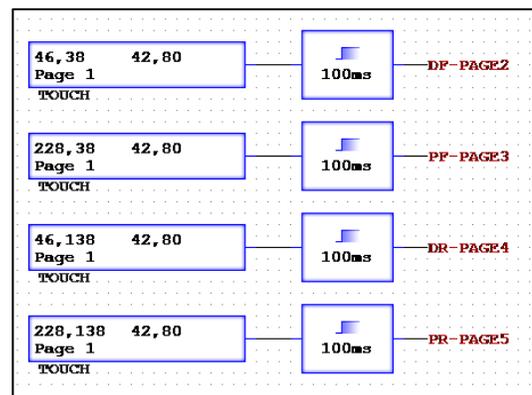
1. Draw X=48 Draw Y=40 Draw Page=1
2. Draw X=230 Draw Y=40 Draw Page=1
3. Draw X=48 Draw Y=140 Draw Page=1
4. Draw X=230 Draw Y=140 Draw Page=1

4. Now create 5 CONSTANTS. Set them as values 1, 2, 3, 4, and 5. Label wires from each as “Page-One”, “Page-Two”, and so on up to “Page-Five”.
5. Insert a PAGE part and label the input wire as “Page-Ctrl”.

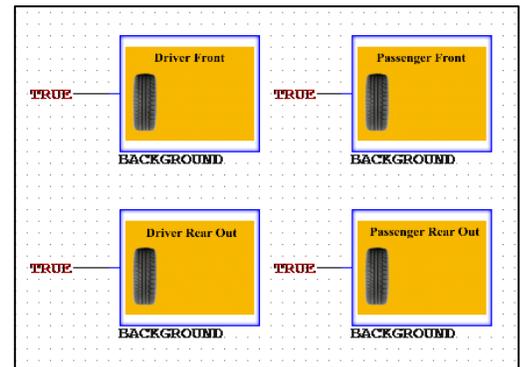


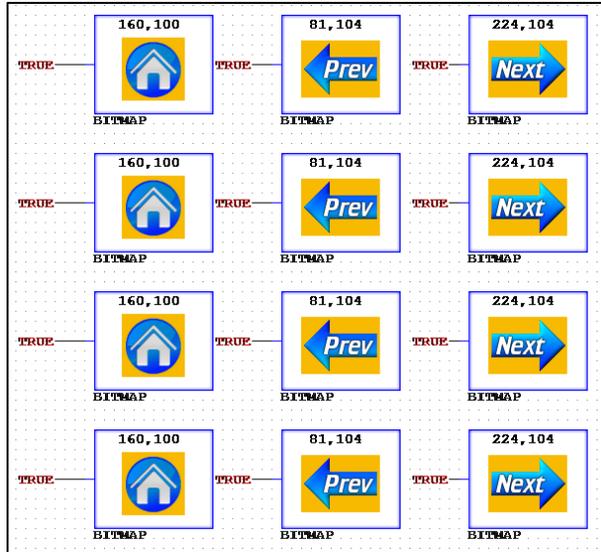
6. After that, insert 4 TOUCH parts and 4 DELAYON parts. Use the following values for each TOUCH part.

1. Draw X=46 Draw Y=38 Width=42 Height=80
2. Draw X=228 Draw Y=38 Width=42 Height=80

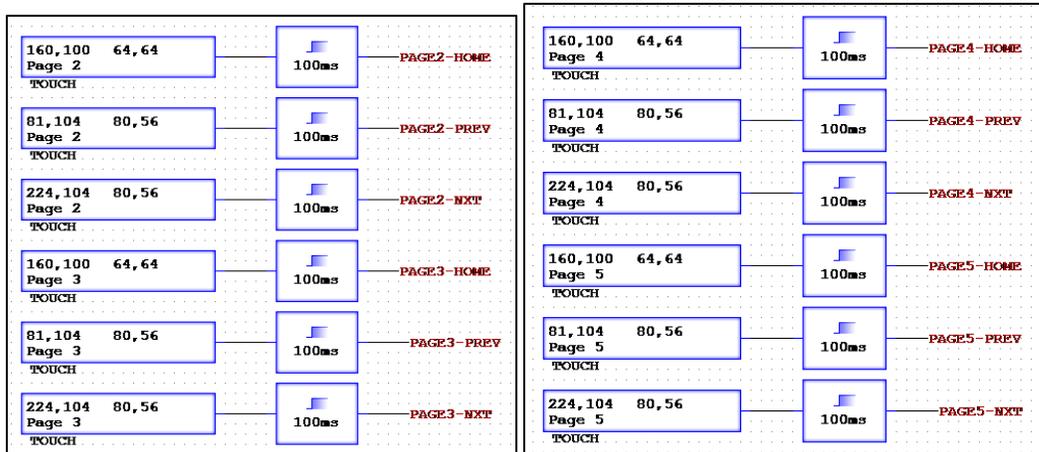


3. Draw X=46 Draw Y=138 Width=42 Height=80
4. Draw X=228 Draw Y=138 Width=42 Height=80
7. Attach a DELAYON to each of the TOUCH outputs and set all the DELAYON timers to 100ms. For the output of the DELAYON with the TOUCH(46,38), label the wire "DF-Page2". For the output of the DELAYON with the TOUCH(228,38), label the wire "PF-Page3". For the output of the DELAYON with the TOUCH(46,138), label the wire "DR-Page4". Finally, for the output of the DELAYON with the TOUCH(228,138), label the wire "PR-Page5".
8. Next is to create the pages for each tire. Insert 4 BACKGROUNDS. Set the first to Draw Page 2 and under User\Manual, Image to "tire1". Set the second to Draw Page 3 and under User\Manual, Image to "tire2". Set the third to Draw Page 4 and under User\Manual, Image to "tire3". Set the first to Draw Page 5 and under User\Manual, Image to "tire6". Set each to TRUE.
9. There are 3 buttons on each of those pages, so 12 BITMAPS need to be inserted. For page 2, set one BITMAP to Draw X=160, Draw Y=100, Draw Page=2, and under User\Manual, Image to "home". Click OK. For the second BITMAP, set Draw X=81, Draw Y=104, Draw Page=2, and under User\Manual, Image to "prev". For the third BITMAP, set Draw X=224, Draw Y=104, Draw Page=2, and under User\Manual, Image to "next".
10. Repeat step 9, using the same locations and images, except change the Draw Page. Repeat for pages 3, 4, and 5. Set all the BITMAPS to TRUE.





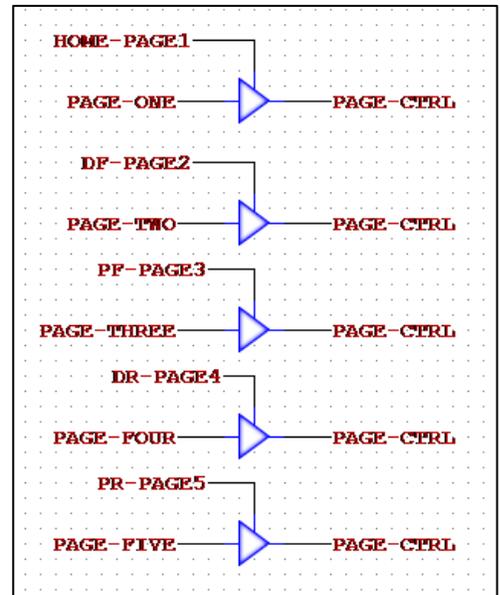
11. Those buttons need TOUCH parts, so insert 12 TOUCH parts with 12 DELAYONs. Set each DELAYON to 100ms and attach it to the output of each TOUCH part.
12. For the TOUCH parts on page 2, set the first one to Draw X=160, Draw Y=100, Draw Page=2, Width=64, and Height=64. Label the wire output from the DELAYON as "Page2-Home". For the second TOUCH part, set Draw X=81, Draw Y=104, Draw Page=2, Width=80, and Height=104. Label the wire output from the DELAYON as "Page2-Prev". For the third TOUCH part, set Draw X=224, Draw Y=104, Draw Page=2, Width=80, and Height=56. Label the wire output from the DELAYON as "Page2-Nxt".
13. Repeat step 12 except change the Draw Page and the output wire label. Do this for pages 3, 4, and 5. Label the outputs the same as page 2 but change the number in the name (see below).



## 7.2 Using Buffers for Pages

14. To create the page navigation, buffers are going to be used. For the main home page, insert 5 BUFFERS. Label each of the output wires to “Page-Ctrl”. For the inputs, label them as “Page-One”, “Page-Two”, and so on. Use the following to know what to set the control wire.

1. Page-One buffer- label control “Home-Page1”
2. Page-Two buffer- label control “DF-Page2”
3. Page-Three buffer- label control “PF-Page3”
4. Page-Four buffer- label control “DR-Page4”
5. Page-Five buffer- label control “PR-Page5”



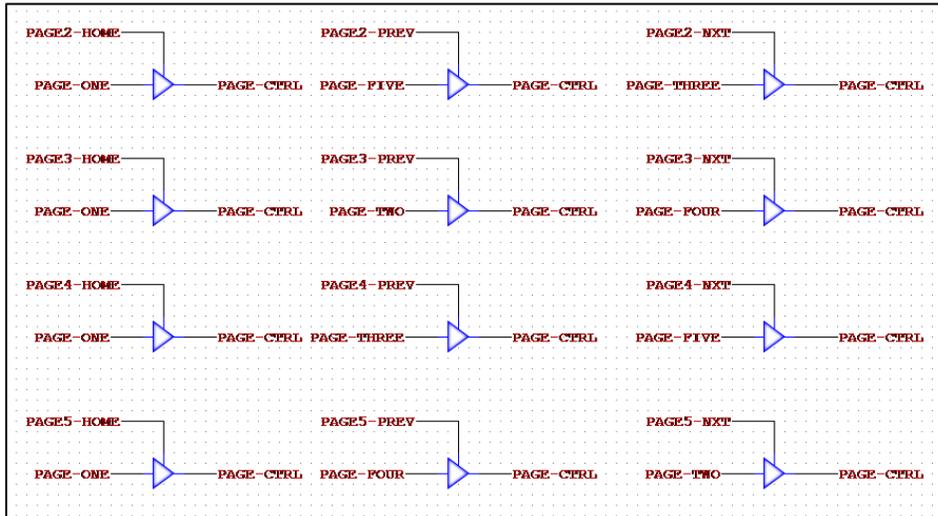
15. To prevent TOUCH area issues from multiple sources to the same wire, each page needs its own logic and wire label names. Insert 12 BUFFERS, 3 will be used for each page.

16. For Page 2, set the output of the 3 BUFFERS to “Page-Ctrl”. For the first BUFFER, set the control wire to “Page2-Home” and the input to “Page-One”. For the second BUFFER, set the control wire to “Page2-Prev” and the input to “Page-Five”. For the final BUFFER, set the control wire to “Page2-Nxt” and the input to “Page-Three”.

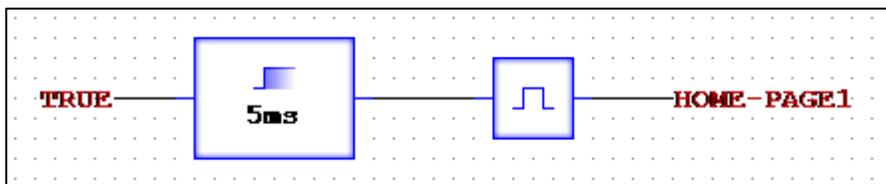
17. For Page 3, set the output of the 3 BUFFERS to “Page-Ctrl”. For the first BUFFER, set the control wire to “Page3-Home” and the input to “Page-One”. For the second BUFFER, set the control wire to “Page3-Prev” and the input to “Page-Two”. For the final BUFFER, set the control wire to “Page3-Nxt” and the input to “Page-Four”.

18. For Page 4, set the output of the 3 BUFFERS to “Page-Ctrl”. For the first BUFFER, set the control wire to “Page4-Home” and the input to “Page-One”. For the second BUFFER, set the control wire to “Page4-Prev” and the input to “Page-Three”. For the final BUFFER, set the control wire to “Page4-Nxt” and the input to “Page-Five”.

19. For Page 5, set the output of the 3 BUFFERS to “Page-Ctrl”. For the first BUFFER, set the control wire to “Page5-Home” and the input to “Page-One”. For the second BUFFER, set the control wire to “Page5-Prev” and the input to “Page-Four”. For the final BUFFER, set the control wire to “Page5-Nxt” and the input to “Page-Two”.

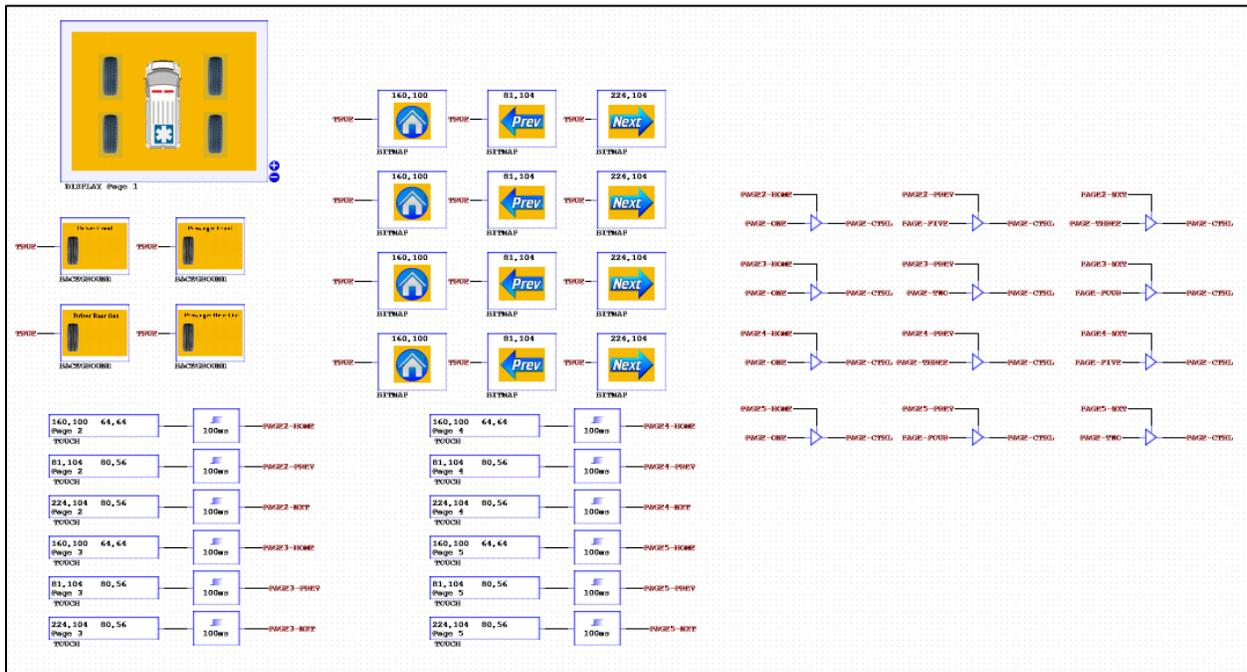
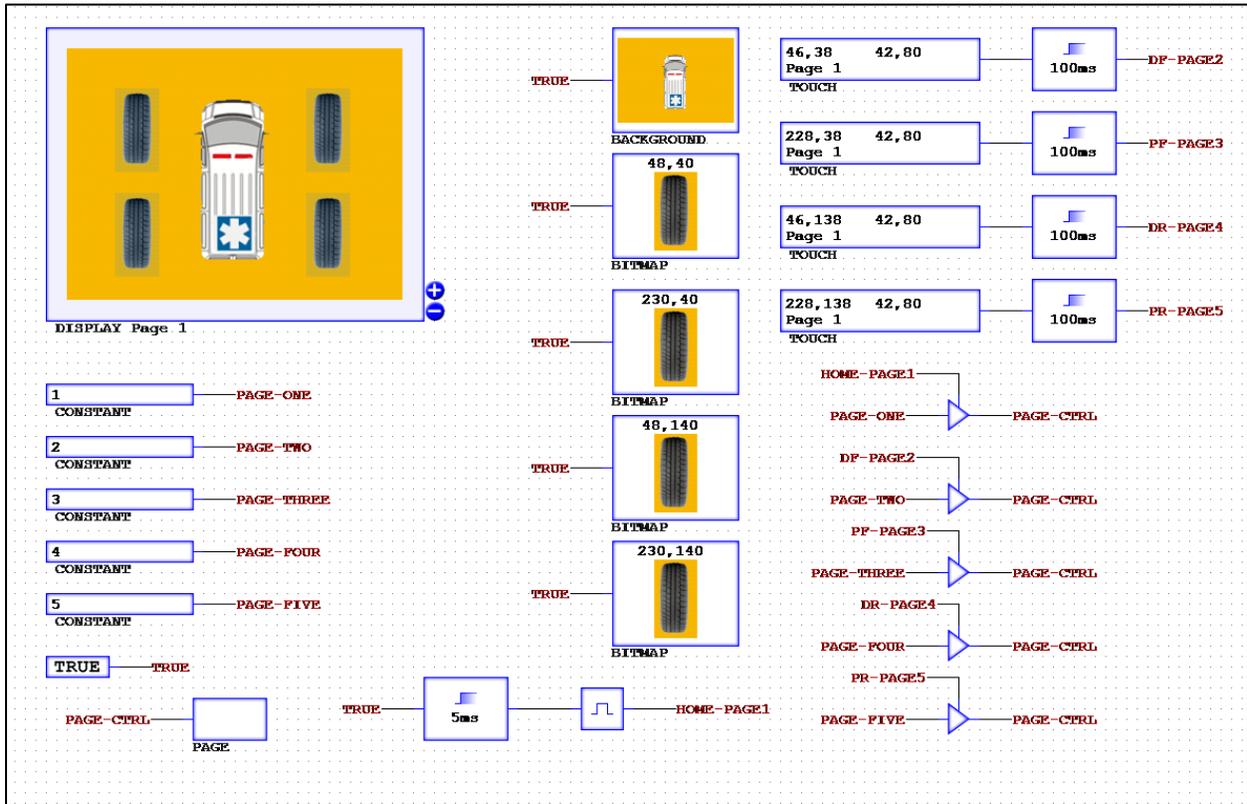


20. The final step is to make sure when the program starts that the page is set to page one. Insert a DELAYON and a PULSE part. Attach a TRUE to the input of the DELAYON and attach the output of the DELAYON to the PULSE input. The output of the PULSE should be “Home-Page1”. Finally, set the timer on the DELAYON to 5ms.



21. Run the simulator, clicking on each tire to go to its individual page and then using the buttons to navigate between. Using buffers creates more ways to navigate pages over just using a counter.

# 7.3 Result



# Chapter 8 – CAN Receiver

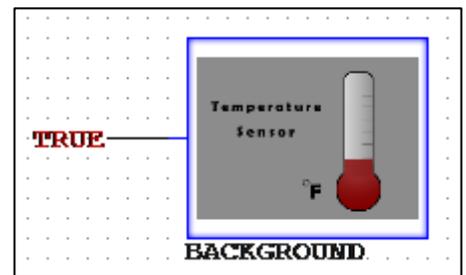
## 8.1 Using CANRX

At the end of this CAN Receiver tutorial, you will be able to use the CANRX part to take in information and process it within the program. Also, learn about how to simulate input values within the software simulator and use the execute part to call other program scripts.

1. Open up a new CANcreate workspace and insert a DISPLAY.

2. Insert a TRUE constant and label the wire “true”.

3. Add in a BACKGROUND part and set the folder to User\Manual and Image to “temp\_sensor”. Click OK. Set to true.



4. Now, add in 6 BITMAP parts. For five of them, set the folder to User\Manual and Image to “temp\_block”. Use the following Draw X and Draw Y instructions for each BITMAP.

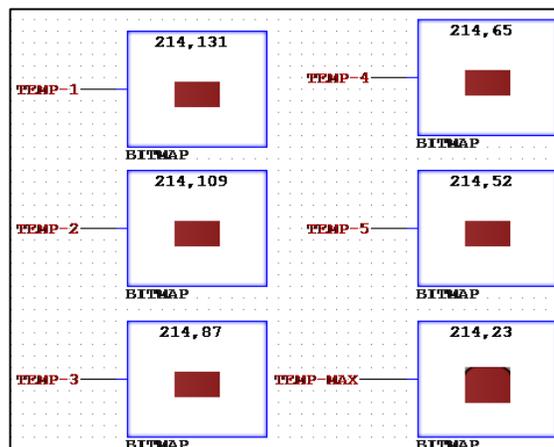
1. BITMAP 1 = Draw X:214 Draw Y:131 Wire Label: “Temp-1”

2. BITMAP 2 = Draw X:214 Draw Y:109 Wire Label: “Temp-2”

3. BITMAP 3 = Draw X:214 Draw Y:87 Wire Label: “Temp-3”

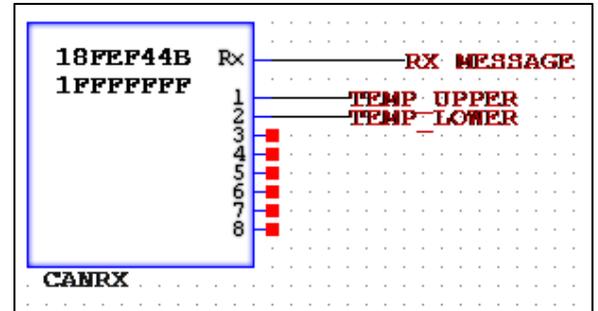
4. BITMAP 4 = Draw X:214 Draw Y:65 Wire Label: “Temp-4”

5. BITMAP 5 = Draw X:214 Draw Y:52 Wire Label: “Temp-5”



5. For the sixth one, set the folder to User\Manual and Image to “temp\_block\_top”. Set Draw X to 214 and Draw Y to 23. Label the wire “Temp-Max”. Click OK.

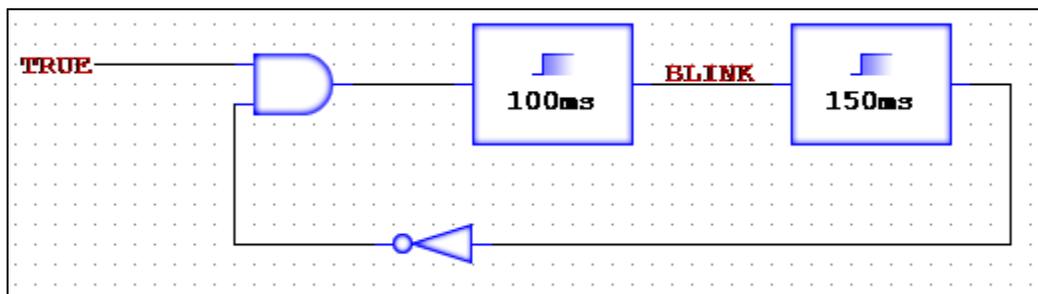
6. Under **Part->Display**, select CANRX. Insert it into the workspace. Under part properties, change the CANID to “18FEF44B” and CAN Mask to “1FFFFFFF”. This will change depending on the CAN device and what information is needed. This is merely an example of what the Hex values would be. Click OK.



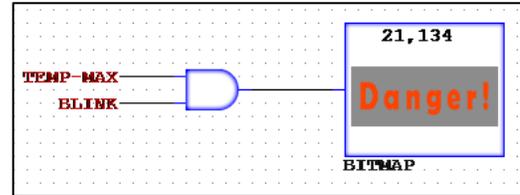
7. Label the RX node, “RX Message”. This is where the full Hex code will be taken in. Then, label the wire from node 1 of the CANRX as “Temp\_Upper” and node 2 as “Temp\_Lower”.

**Note:** Node 1 is the first Byte of the data, Node 2 is the second Byte of the data and so on.

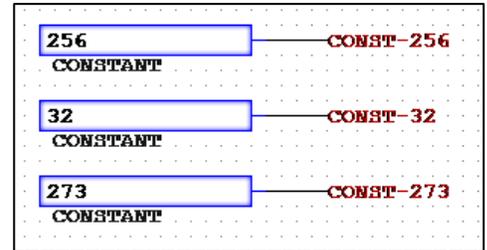
8. Next, create a blinking circuit like in chapter 4. An AND2, two DELAYONs, and a NOT gate are needed. Attach a TRUE to one of the AND2 inputs and the output of the NOT gate to the second input. Wire the output of the AND2 gate to a DELAYON input. Connect the output of that DELAYON to the input of the NOT gate. Then connect the output of that DELAYON to the input of the other DELAYON. Make the first DELAYON 100ms and the second DELAYON 150ms. Label the wire between them “Blink”.



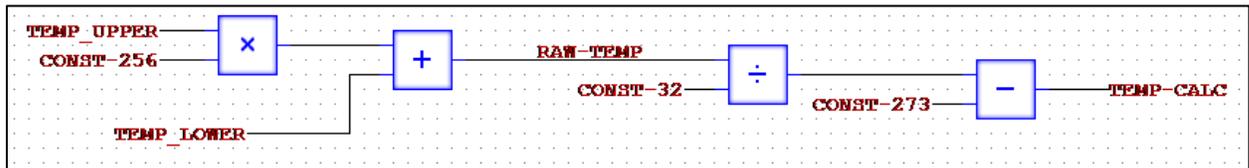
9. Now, insert an AND2 gate and a BITMAP. Set the first AND2 input as “Temp-Max” and the second as “Blink”. Wire the AND2 output to the BITMAP. Under the BITMAP properties, set the Draw X to 21 and Draw Y to 134. Change the Folder to User\Manual and Image to “temp\_warning”. Click OK.



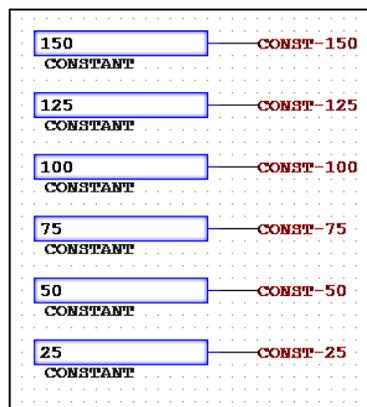
10. To decode the Hex input values, a logic circuit is needed. Insert 3 constants, set values as 32, 256, and 273. Label their wires “Const-32”, “Const-256”, and “Const-273”.



11. Add in a MULTIPLY, ADD, DIVIDE, and SUBTRACT. Start with the MULTIPLY, setting the inputs as “Temp\_Upper” and “Const-256”. Connect that output to the ADD and the second ADD input to “Temp\_Lower”. Connect that output to the DIVIDE and the second DIVIDE input to “Const-32”. Finally, connect that output to the SUBTRACT and the second input to “Const-273”. The SUBTRACT output should be “Temp-Calc”. (This is a random scaling factor that allows for a 16bit number to be expressed with the software)

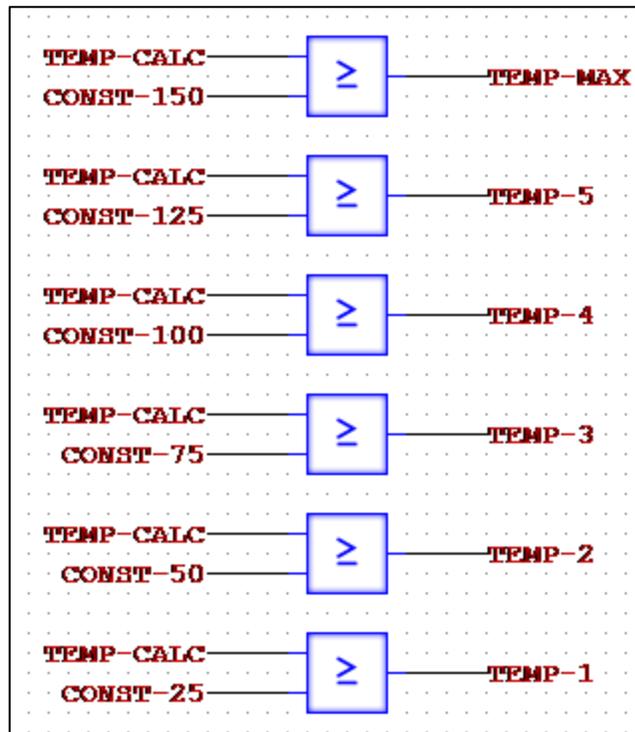


12. Insert 6 more constants, using the values 25, 50, 75, 100, 125, and 150. Label them like earlier, such as “Const-25” and so on.

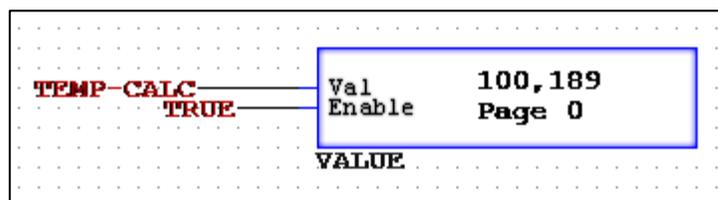


13. Next, insert 6 GREATEREQUAL parts. Use the following to label each one correctly.

1. Input 1:"Temp-Calc" Input 2:"Const-150" Output:"Temp-Max"
2. Input 1:"Temp-Calc" Input 2:"Const-125" Output:"Temp-5"
3. Input 1:"Temp-Calc" Input 2:"Const-100" Output:"Temp-4"
4. Input 1:"Temp-Calc" Input 2:"Const-75" Output:"Temp-3"
5. Input 1:"Temp-Calc" Input 2:"Const-50" Output:"Temp-2"
6. Input 1:"Temp-Calc" Input 2:"Const-25" Output:"Temp-1"

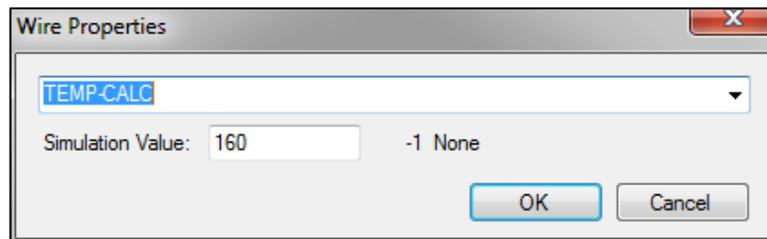


14. Finally, add in a VALUE part. Set Draw X to 100 and Draw Y to 189. Change the Foreground to (0,0,0) and Background to (140,140,140). Click OK. Set the Val node to "Temp-Calc" and the Enable to TRUE.



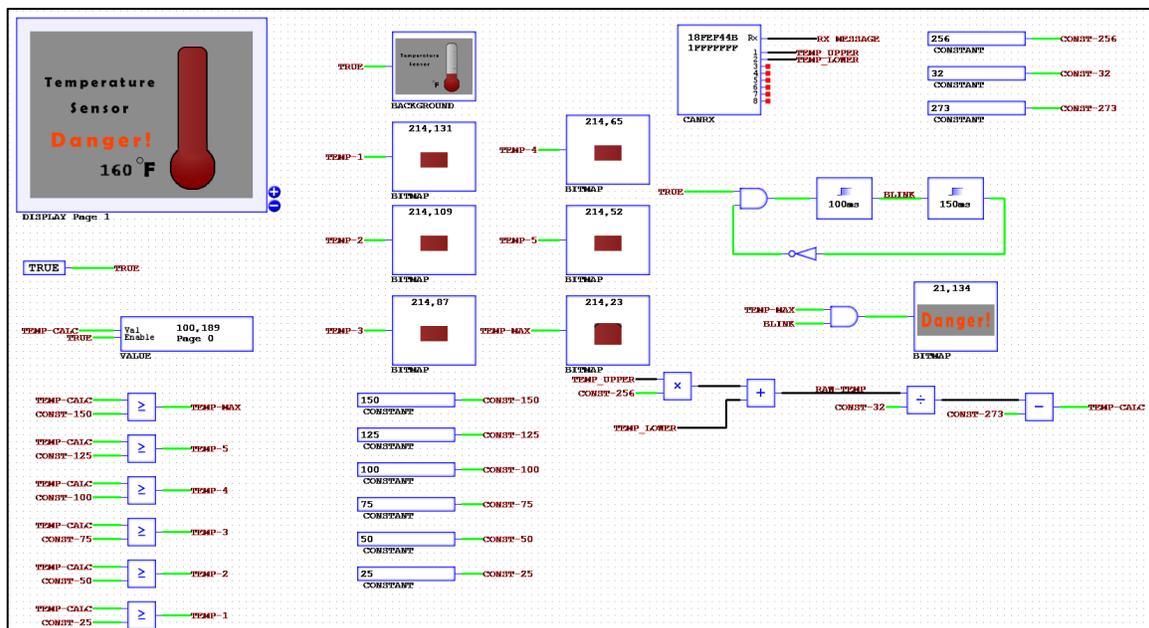
## 8.2 simulation values

15. Find the wire after the SUBTRACT that is labeled “Temp-Calc”. Double-click the wire. There is a box titled “Simulation Value”. The default value is 0 but change it to 160. Click OK. This enables the simulation to test certain values coming from the CANRX potentially when physically running. The 160 representing a degree value in Fahrenheit. **Note:** This only works if the program has not been routed. If routed, the values on the CANRX are the ones needed to be edited to see changes in the simulation.



16. Run the simulator. The value 160 should be displayed on the screen, with the thermometer at max and “Danger” blinking on the screen. Other values can be tested by changing the simulation value. Look at the results of changing the value to 12, 45, 67, 82, and 123.

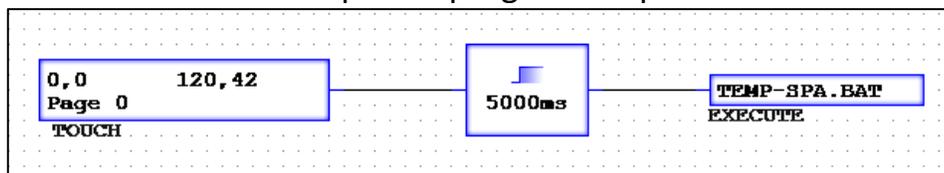
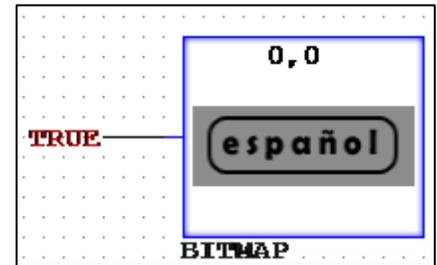
## 8.3 Result



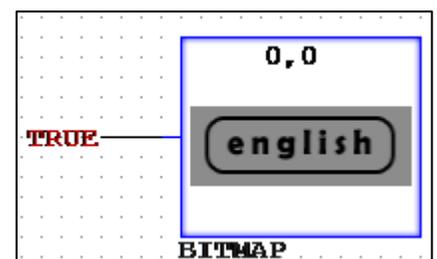
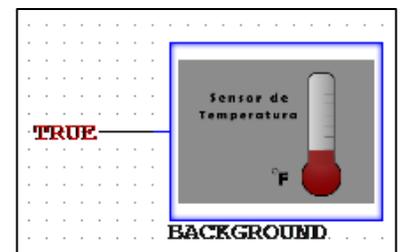
## 8.4 Extension: Execution Uses

The execution part can be used not only to run the set clock and calendar functions, but to run other program scripts. A good use of this is the need to have multiple language support. Using the previously completed Temperature Sensor, complete these steps to make a Spanish compatible version.

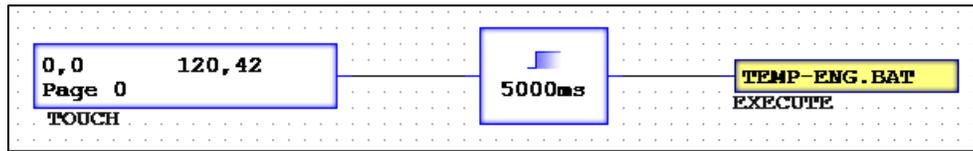
1. Open the Temperature Sensor project and resave it with “\_En”.  
Ex: Therm\_En.ccs
2. First is to add in another BITMAP. Set it to TRUE and leave Draw X/Y equal to 0. Change Folder to User\Manual and Image to “temp\_spanish”. Click OK.
3. Next, insert a TOUCH part. Set Draw X and Draw Y to 0. Make the Width 120 and the Height 42. Click OK. Wire a DELAYON to the TOUCH part. Set the timer to 5000ms (5 seconds). Finally, wire an EXECUTE part to the output of the DELAYON. Under Filename, type “Temp-Spa.bat”. This will be the batch file that runs the Spanish program script.



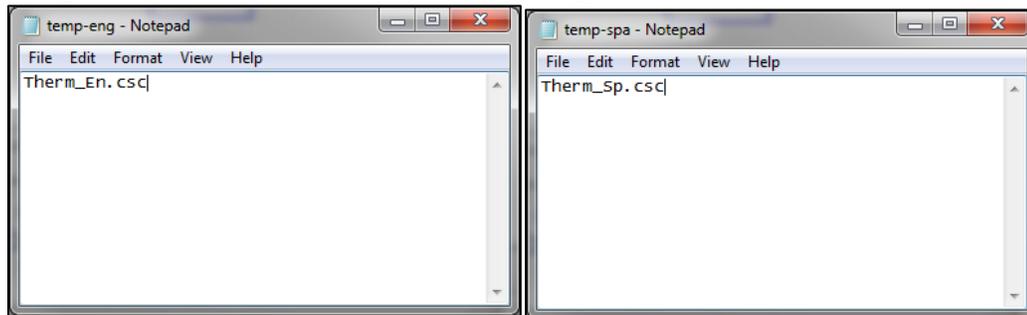
4. Now copy all the contents of the workspace and open a new blank file. Paste the contents and save this new workspace with “\_Sp”.  
Ex: Therm\_Sp.ccs
5. Go to the BACKGROUND properties and change the Image to “temp\_sensor\_spanish”. Click OK.
6. Locate the BITMAP at (0,0) and go to part properties. Change the Image to “temp\_english”. Click OK.
7. Next find the “Danger” BITMAP. Under part properties, change Draw X to 14 and Draw Y to 121. Change the Image to “temp\_spanish\_warning”. Click OK.



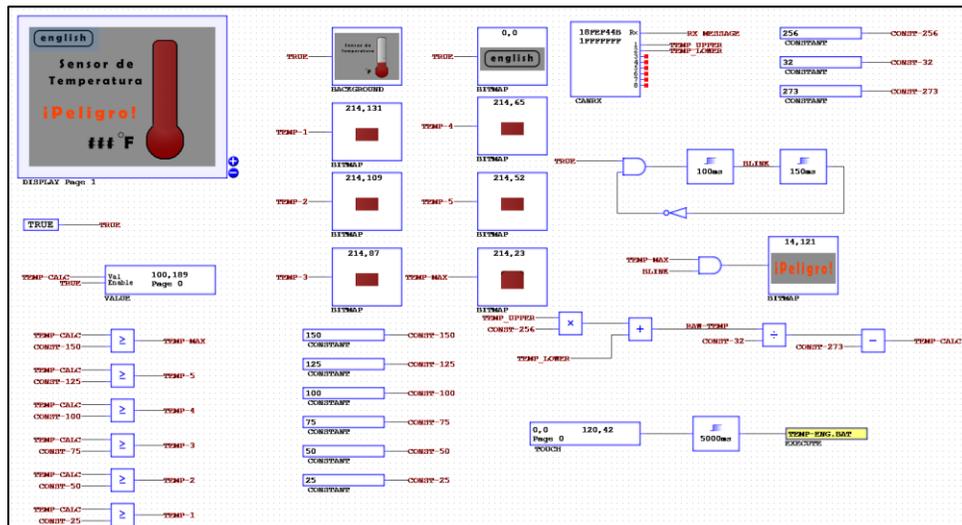
- Finally, find the EXECUTE part and go to properties. Change the Filename to "Temp-Eng.bat". Click OK.



- To make the batch files, open up a blank Notepad window. Type the name of your Spanish program ".csc" and save as "Temp-Spa.bat". Open up a second blank Notepad window. Type the name of your English program ".csc" and save as "Temp-Eng.bat".



- Run the simulator for the Spanish version and it will run just like the English one. The batch files can't be tested with the program simulator but holding either the "English" or "Spanish" button for 5 seconds will launch the other script. Install the scripts onto the device to see them in action.

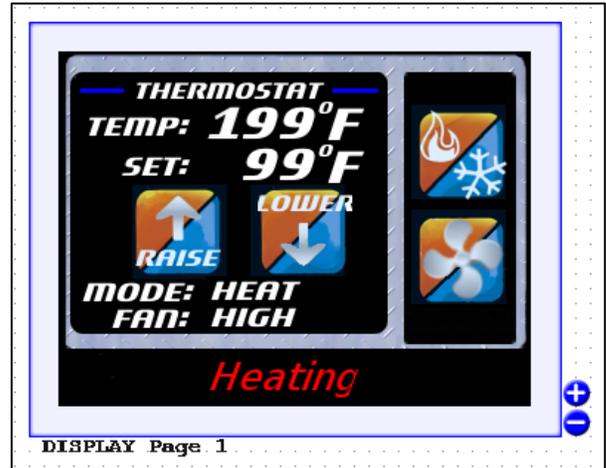


# Chapter 9 – CAN Transmitter

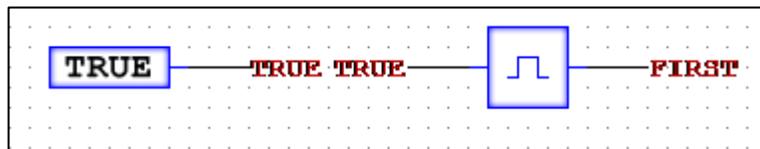
## 9.1 Using Personal Font

At the end of this CAN Transmitter tutorial, you will be able to create and use your own font to simulate a thermostat. Also, use the CANTX part to transmit data to an external source to use.

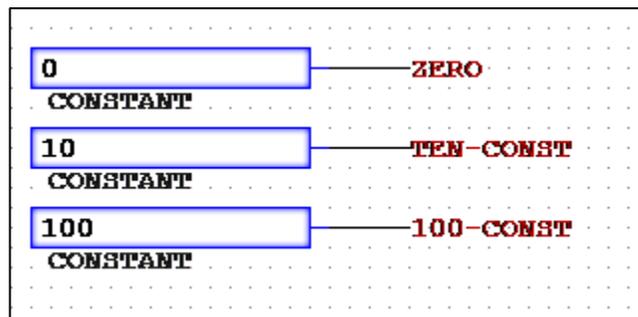
1. Open up a new CANcreate workspace and insert a DISPLAY.
2. Insert a TRUE part and label the wire “True”. Place a BACKGROUND into the workspace, set it equal to TRUE, and under properties set Folder to User\Manual. Then set the Image to “thermo-controller”. Click OK.



3. Next, insert a PULSE. Labeling the input “True” and the output as “First”

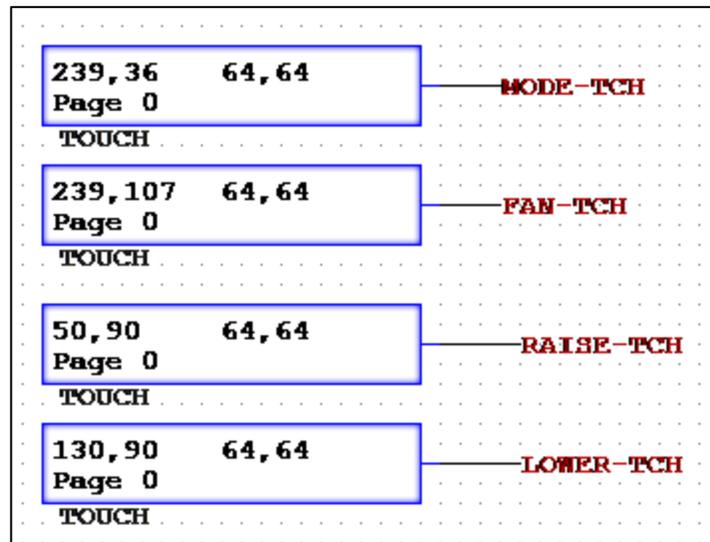


4. Now insert 3 CONSTANT parts. Use the values 0, 10, and 100. Label them as “Zero”, “Ten-Const”, and “100-Const”.

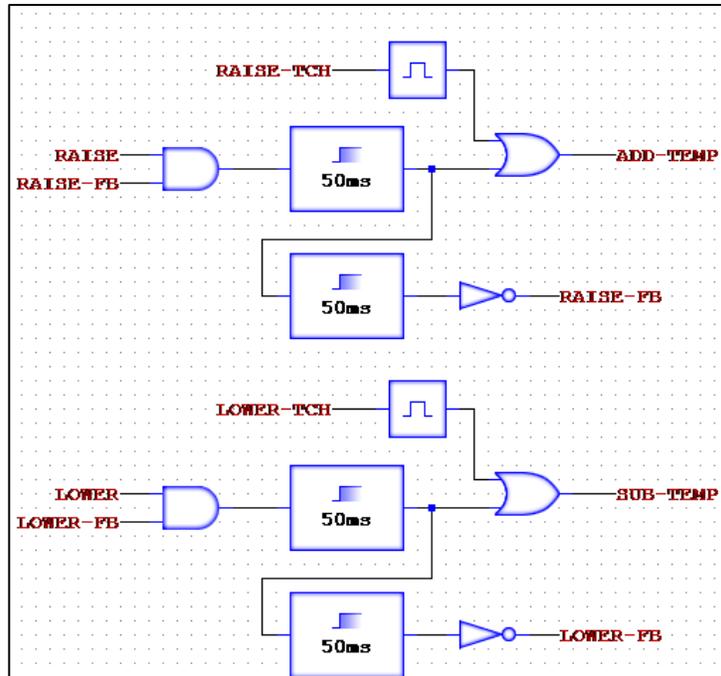


5. Place two TOUCH parts into the workspace. Label the output of one to “Mode-TCH” and the output of the other to “Fan-TCH”. Under properties for the “Mode-TCH” TOUCH part, set Draw X to 239, Draw Y to 36, Width to

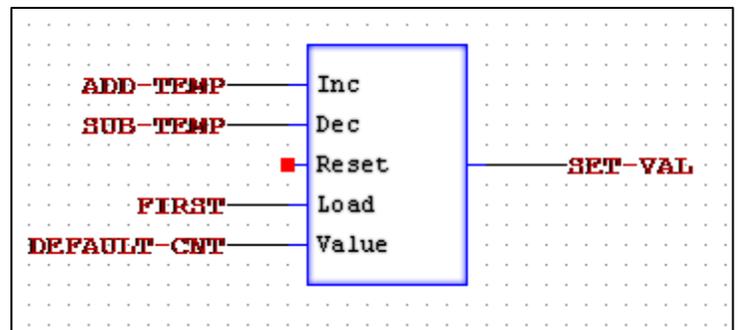
64, and Height to 64. Click OK. For the second TOUCH part, set Draw X to 239, Draw Y to 107, Width to 64, and Height to 64. Click OK.



6. Insert 2 more TOUCH parts. Label the output of one to “Raise-TCH” and output of the other “Lower-TCH”. On the TOUCH with output “Raise-TCH”, set Draw X to 50, Draw Y to 90, Width to 64, and Height to 64. Click OK. On the second TOUCH part, set Draw X to 130, Draw Y to 90, Width to 64, and Height to 64. Click OK.
7. Now an increase temperature circuit must be made. Insert an AND2, two DELAYONs, a PULSE, a NOT, and an OR2 part. Label the two inputs on the AND2, “Raise” and “Raise-FB”. Connect the output of the AND2 to one of the DELAYON parts. Change the DELAYON properties to a 50ms timer. Branch two wires from the output of the DELAYON to the input of the OR2 and the second DELAYON part. Change the timer on the second DELAYON to 50ms, too. Wire the output of the DELAYON to the NOT gate and label its output “Raise-FB”. Locate the PULSE part and label its input as “Raise-TCH”. Next wire the output of the PULSE to the other input on the OR2 gate. Finally, label the output of the OR2 as “Add-Temp”.
8. Repeat step 7 to create a decrease temperature circuit. Change any label that was “Raise-TCH” before to “Lower-TCH”. Also, change any label that was “Raise-FB” to “Lower-FB”. Finally, change “Raise” to “Lower” and the output of the OR2 to “Sub-Temp”.



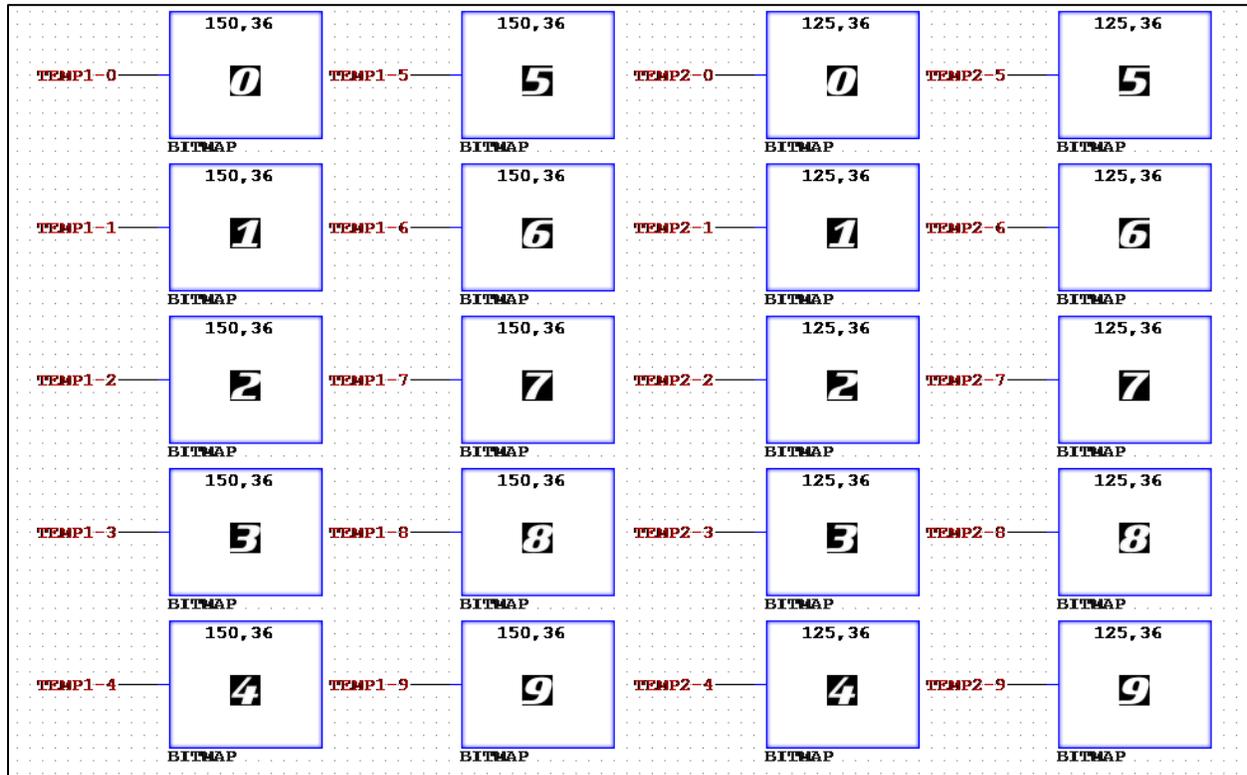
9. Insert a COUNTER. Label the Inc to “Add-Temp”, the Dec to “Sub-Temp”, the Load to “First”, and the Value to “Default-Cnt”. Change the output to “Set-Val”. Under COUNTER properties, change the Range to 200 and leave Disable Overflow unchecked. Click OK.



10. To use your own font for the temperature readout, each number must be imported as a BITMAP. Insert 10 BITMAPS. Each one will have Draw X = 150 and Draw Y = 36. Use the following Images and labels for each. (Folder is User\Manual)

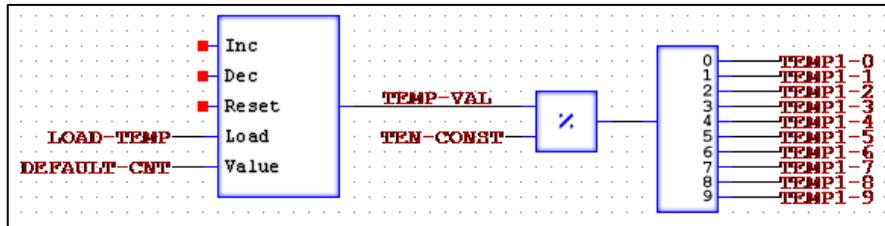
1. Label: “Temp1-0” Image: “numeral0”
2. Label: “Temp1-1” Image: “numeral1”
3. Label: “Temp1-2” Image: “numeral2”
4. Label: “Temp1-3” Image: “numeral3”
5. Label: “Temp1-4” Image: “numeral4”

6. Label: "Temp1-5" Image:"numeral5"
7. Label: "Temp1-6" Image:"numeral6"
8. Label: "Temp1-7" Image:"numeral7"
9. Label: "Temp1-8" Image:"numeral8"
10. Label: "Temp1-9" Image:"numeral9"

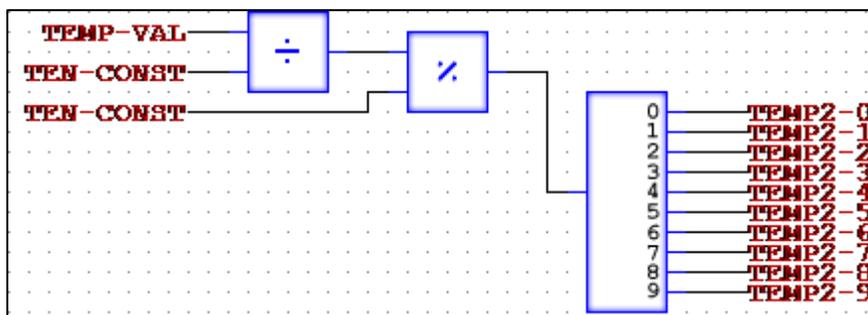


11. Another set of numbers will be needed, so insert 10 more BITMAPs. Use Draw X = 125 and Draw Y=36 for all. Repeat step 10 except use the naming convention "Temp2-0", "Temp2-1" and so on to "Temp2-9".

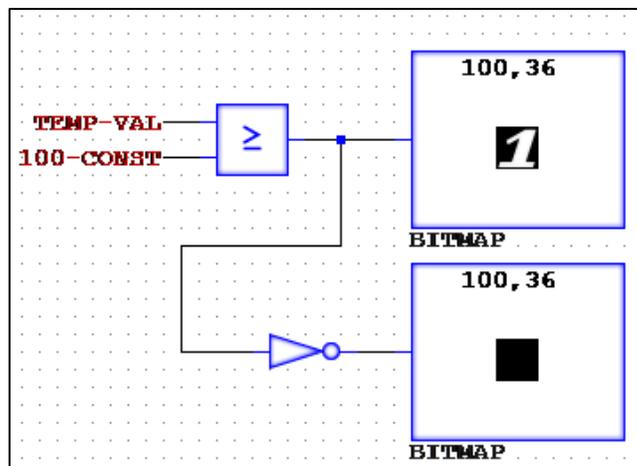
12. Now insert a COUNTER. Under COUNTER properties, change the Range to 200 and leave Disable Overflow unchecked. Click OK. Label Load as "Load-Temp" and Value as "Default-CNT". Insert a MODULUS part and connect it to the output of the COUNTER. Label that wire as "Temp-Val". Label the bottom input of the MODULUS as "Ten-Const". Place a DECODE part next to the MODULUS and wire them together. Label the DECODE outputs as (starting from the top) "Temp1-0", "Temp1-1", and so on until "Temp1-9".



13. Next, insert a DIVIDE, a MODULUS, and a DECODE. Label the DIVIDE inputs (top to bottom) as “Temp-Val” and “Ten-Const”. Connect the output to the top input of the MODULUS. For the bottom input, label it as “Ten-Const”. Wire the MODULUS output to the DECODE. Label the DECODE outputs as (starting from the top) “Temp2-0”, “Temp2-1”, and so on until “Temp2-9”.

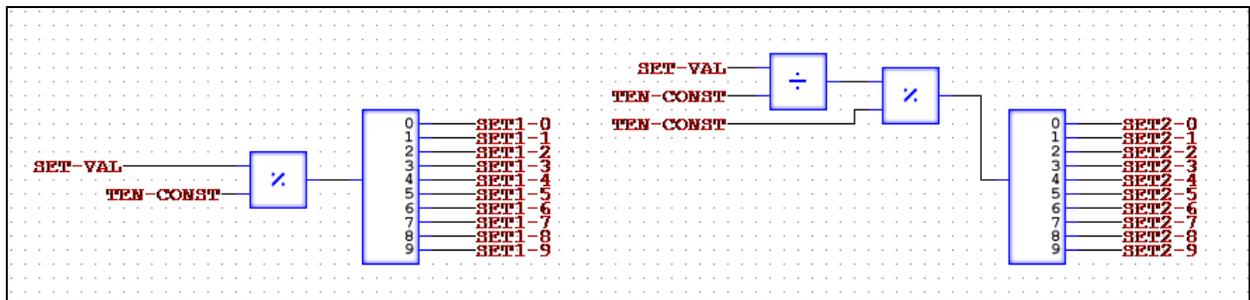


14. Insert a GREATEREQUAL, a NOT, and 2 BITMAPS. Label the GREATEREQUAL (top to bottom) as “Temp-Val” and “100-Const”. Branch two wires from the GREATEREQUAL output, connecting to one of the BITMAPS and the to the NOT. The output of the NOT should attach to the second BITMAP. For the BITMAP without the NOT before it, set Draw X to 100 and Draw Y to 36. Set Folder to User\Manual and Image to “numeral1”. Click OK. On the second BITMAP, set Draw X to 100 and Draw Y to 36. Change Folder to User\Manual and Image to “Blank24”. Click OK.



15. The set section needs similar circuits as well. Insert a MODULUS and DECODE part. Label the MODULUS inputs (top to bottom) as “Set-Val” and “Ten-Const”. Connect the output to the DECODE part. Label the DECODE outputs as (starting from the top) “Set1-0”, “Set1-1”, and so on until “Set1-9”.

16. Next, insert a DIVIDE, a MODULUS, and a DECODE. Label the DIVIDE inputs (top to bottom) as “Set-Val” and “Ten-Const”. Connect the output to the top input of the MODULUS. For the bottom input, label it as “Ten-Const”. Wire the MODULUS output to the DECODE. Label the DECODE outputs as (starting from the top) “Set2-0”, “Set2-1”, and so on until “Set2-9”.

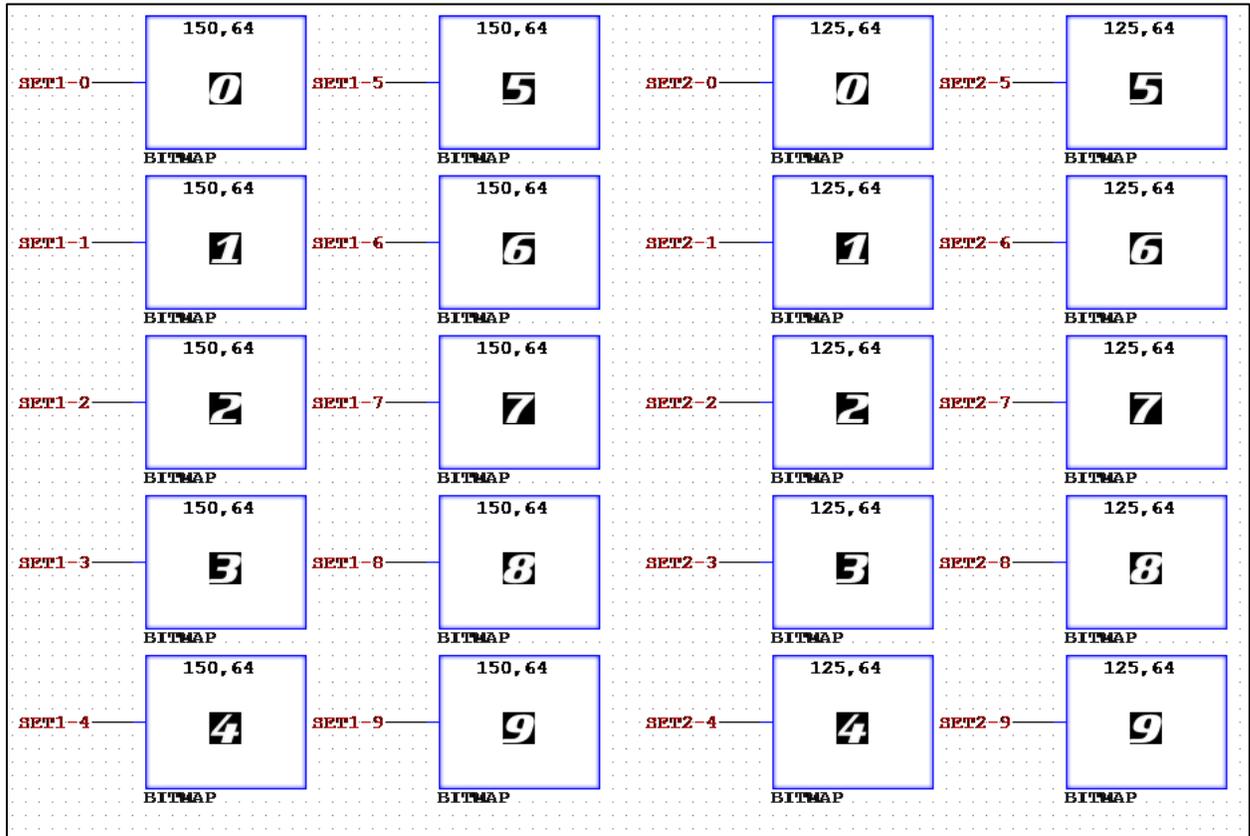


17. To use your own font for the set numbers, each number must be imported as a BITMAP, too. Insert 10 BITMAPS. Each one will have Draw X = 150 and Draw Y = 64. Use the following Images and labels for each. (Folder is User\Manual)

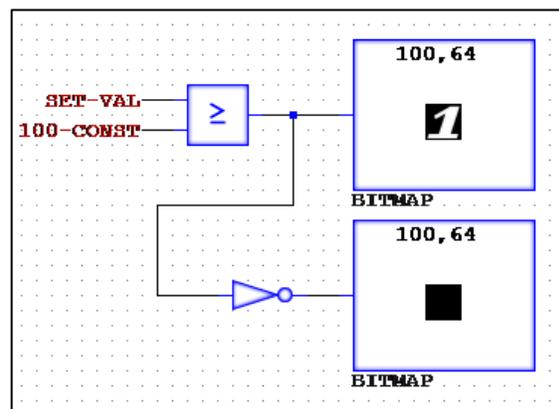
1. Label: “Set1-0” Image:”numeral0”
2. Label: “Set1-1” Image:”numeral1”
3. Label: “Set1-2” Image:”numeral2”
4. Label: “Set1-3” Image:”numeral3”
5. Label: “Set1-4” Image:”numeral4”
6. Label: “Set1-5” Image:”numeral5”
7. Label: “Set1-6” Image:”numeral6”
8. Label: “Set1-7” Image:”numeral7”
9. Label: “Set1-8” Image:”numeral8”

10.Label: “Set1-9” Image:”numeral9”

18.Another set of numbers will be needed, so insert 10 more BITMAPs. Use Draw X = 125 and Draw Y=64 for all. Repeat step 17 except use the naming convention “Set2-0”, “Set2-1” and so on to “Set2-9”.

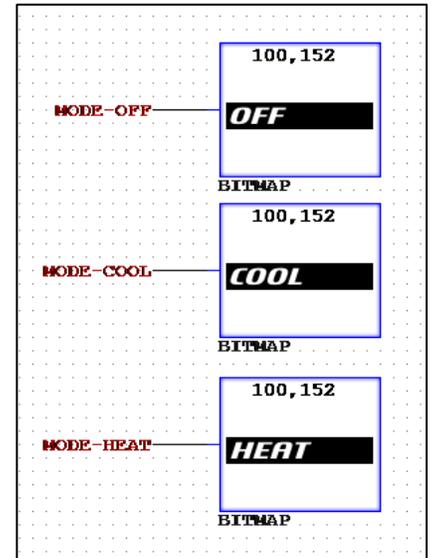


19.Insert a GREATEREQUAL, a NOT, and 2 BITMAPs. Label the GREATEREQUAL (top to bottom) as “Set-Val” and “100-Const”. Branch two wires from the GREATEREQUAL output, connecting to one of the BITMAPs and the to the NOT. The output of the NOT should attach to the second BITMAP. For the BITMAP without the NOT before it, set Draw X to 100 and Draw Y to 64. Set Folder to User\Manual and Image to “numeral1”. Click OK. On the second BITMAP, set Draw X to 100 and Draw Y to 64. Change Folder to User\Manual and Image to “Blank24”. Click OK.

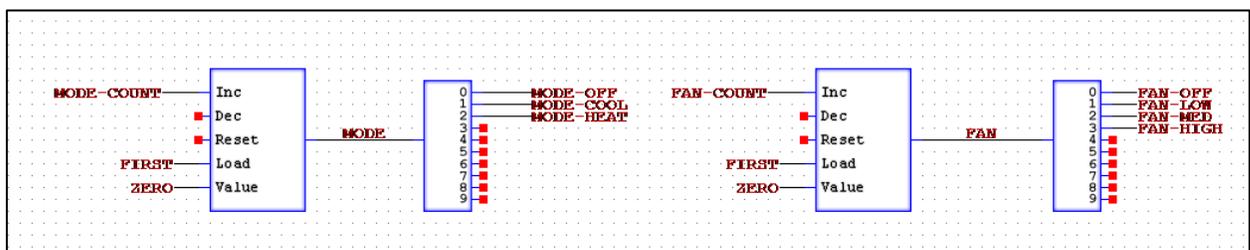


20. To change the Mode on the Thermostat, a COUNTER and a DECODE is needed. Insert those, and label the COUNTER Inc as “Mode-Count”, Load as “First”, and Value as “Zero”. Under COUNTER properties, change the Range to 2 and leave Disable Overflow unchecked. Click OK. Wire the COUNTER to the DECODE, labeling the wire as “Mode”. For the DECODE, label the wires (top to bottom) as “Mode-Off”, “Mode-Cool”, and “Mode-Heat”. (Use nodes 0, 1, and 2)

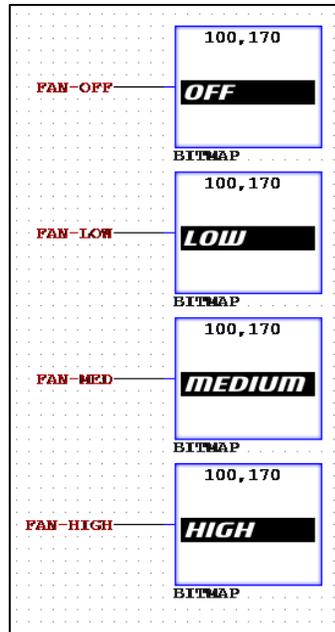
21. Place 3 BITMAPs into the workspace. Label them as “Mode-Off”, “Mode-Cool”, and “Mode-Heat”. For “Mode-Off”, set Draw X to 100, Draw Y to 152, Folder to User\Manual, and Image to “mode1”. Click OK. For “Mode-Cool”, set Draw X to 100, Draw Y to 152, Folder to User\Manual, and Image to “mode2”. Click OK. For “Mode-Heat”, set Draw X to 100, Draw Y to 152, Folder to User\Manual, and Image to “mode3”. Click OK.



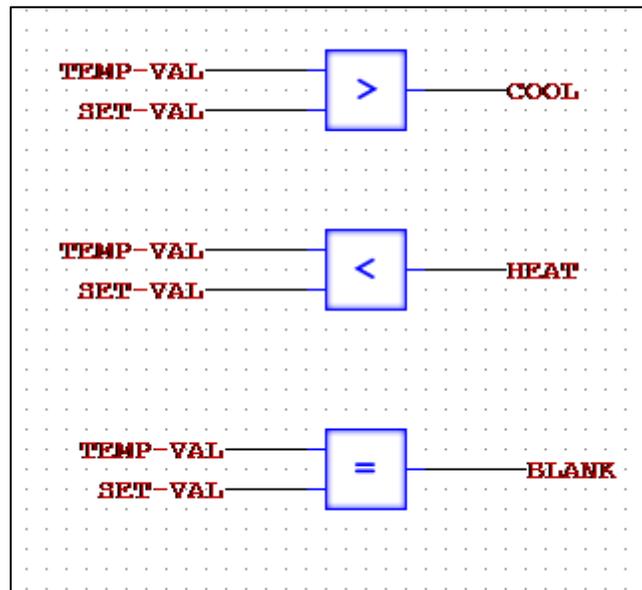
22. To change the Fan on the Thermostat, a COUNTER and a DECODE is needed. Insert those, and label the COUNTER Inc as “Fan-Count”, Load as “First”, and Value as “Zero”. Under COUNTER properties, change the Range to 3 and leave Disable Overflow unchecked. Click OK. Wire the COUNTER to the DECODE, labeling the wire as “Fan”. For the DECODE, label the wires (top to bottom) as “Fan-Off”, “Fan-Low”, “Fan-Med”, and “Fan-High”. (Use nodes 0, 1, 2, and 3)



23. Insert 4 BITMAPs. Label them as “Fan-Off”, “Fan-Low”, “Fan-Med”, and “Fan-High”. Set each Draw X to 100 and Draw Y to 170. Change each Folder to User\Manual. For the “Fan-Off” Image, use “fan1”. For the “Fan-Low” Image, use “fan2”. For the “Fan-Med” Image, use “fan3”. Finally, for the “Fan-High” Image, use “fan4”.

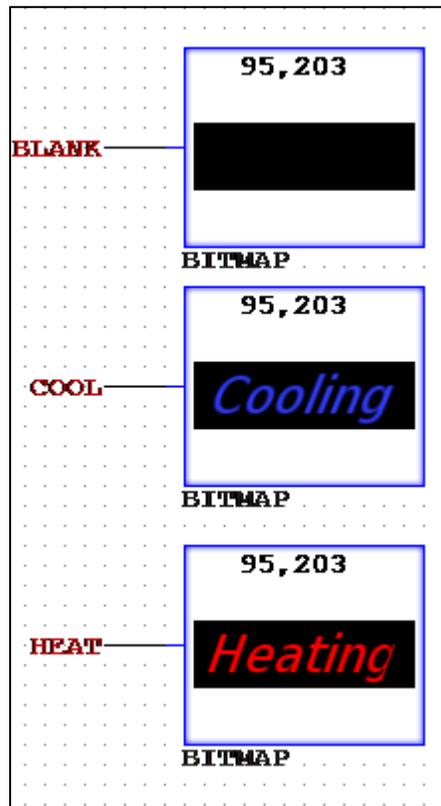


24. To have the thermostat display what action it is performing, the following is needed. Insert a GREATER, a LESS, and an EQUAL part. Label the inputs of the GREATER (top to bottom) as “Temp-Val” and “Set-Val”. The output should be “Cool”. Label the inputs of the LESS (top to bottom) as “Temp-Val” and “Set-Val”. The output should be “Heat”. Label the EQUAL inputs (top to bottom) as “Temp-Val” and “Set-Val”. The output should be “Blank”.



25. Finally, insert 3 BITMAPs. Label them as “Blank”, “Cool”, and “Heat”. For each, Draw X = 95, Draw Y = 203, and Folder is User\Manual. For “Blank”,

set Image to “thermo\_blank”. Click OK. For “Cool”, set Image to “Cooling”. Click OK. For “Heat”, set Image to “heating”. Click OK.



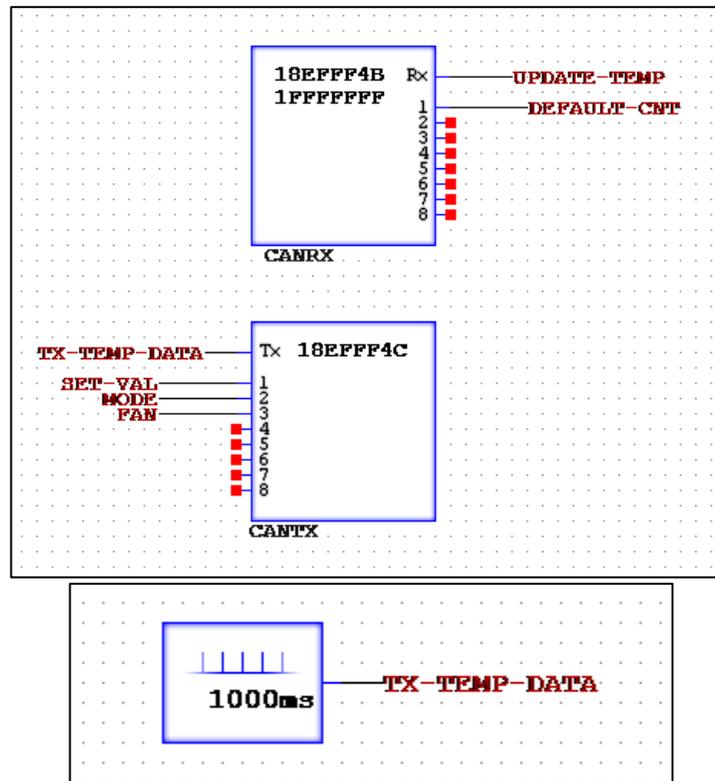
## 9.2 Using CANTX

---

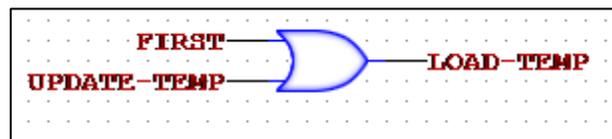
26.To communicate to the CAN system, insert a CANRX and a CANTX. Go to properties of the CANRX and set the CAN ID (which would be the ID of the device that it will be communicating with). For this example, set it to “18EFFF4B” and the CAN Mask to “1FFFFFFF”. Click OK. Set the CAN ID on the CANTX to “18EFFF4C” and use the same CAN Mask from the CANRX. Click OK. (Assumptions that there will be no negative numbers or ones over value 199).

27.For the RX node on the CANRX, label it “Update-Temp” and node 1 as “Default-CNT”. For the TX node on the CANTX, label it as “TX-Temp-Data”.

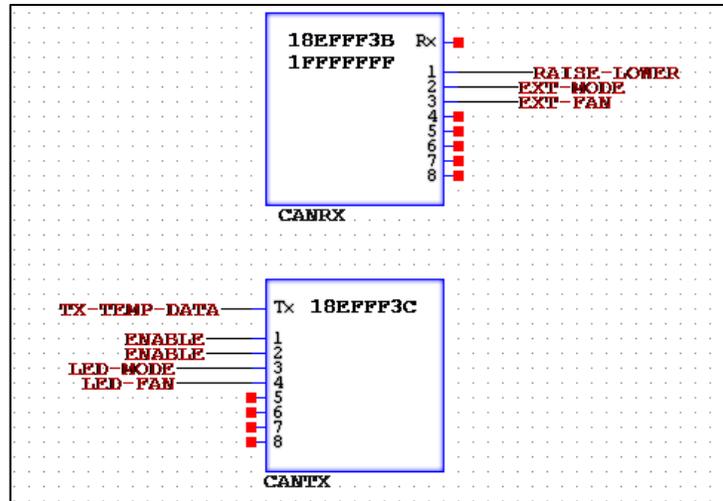
For node 1, label it “Set-Val”, set node 2 as “mode”, and node 3 as “Fan”. Now, insert a TIMER and set the Timer to 1000ms.



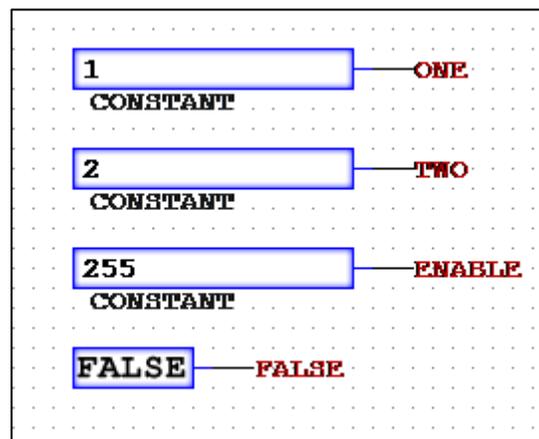
Label the output as “TX-Temp-Data”. This will then send data ever second (1000ms) over the CAN connection. Insert an OR2 part, label the inputs as “First” and “Update-Temp”. Label the output as “Load-Temp”. This will update the temperature when a new data is received.



28. Insert a second part of CANRX and CANTX. For the CANRX, set the CAN ID to “18EFFF3B” and CAN Mask to “1FFFFFFF”. Label node 1 as “Raise-Lower”, node 2 as “Ext-Mode”, and node 3 as “Ext-Fan”. For the CANTX, set the CAN ID to “18EFFF3C”. Label the TX node as “Tx-Temp-Data”, node 1 as “Enable”, node 2 as “Enable”, node 3 as “LED-Mode”, and node 4 as “LED-Fan”.



29. Add in 3 CONSTANT parts. Set them as values 1, 2, and 255. Label them "one", "two", and "Enable". Also, insert a FALSE and label it "False".



30. To decode the receiving message, some calculations are needed. Insert 4 EQUAL parts. Use the following labels:

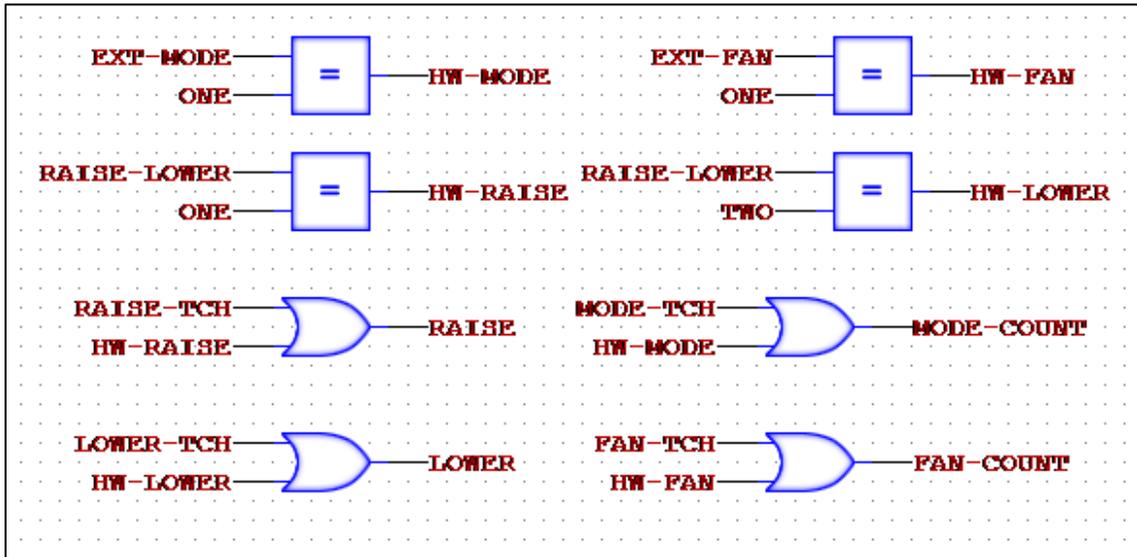
1. Input 1: "Ext-Mode" Input 2: "One" Output: "Hw-Mode"
2. Input 1: "Ext-Fan" Input 2: "One" Output: "Hw-Fan"
3. Input 1: "Raise-Lower" Input 2: "One" Output: "Hw-Raise"
4. Input 1: "Raise-Lower" Input 2: "Two" Output: "Hw-lower"

31. Next insert 4 OR2 parts. Label them using the following:

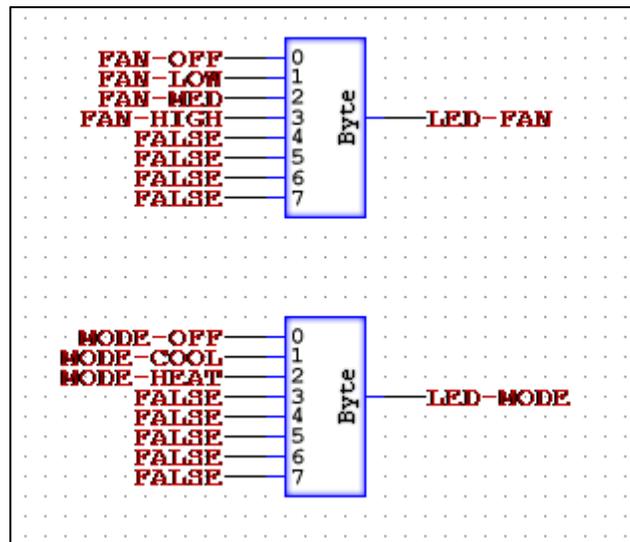
1. Input 1: "Raise-TCH" Input 2: "Hw-Raise" Output: "Raise"
2. Input 1: "Mode-TCH" Input 2: "Hw-Mode" Output: "Mode-Count"

3. Input 1: "Lower-TCH" Input 2: "Hw-Lower" Output: "Lower"

4. Input 1: "Fan-TCH" Input 2: "Hw-Fan" Output: "Fan-Count"



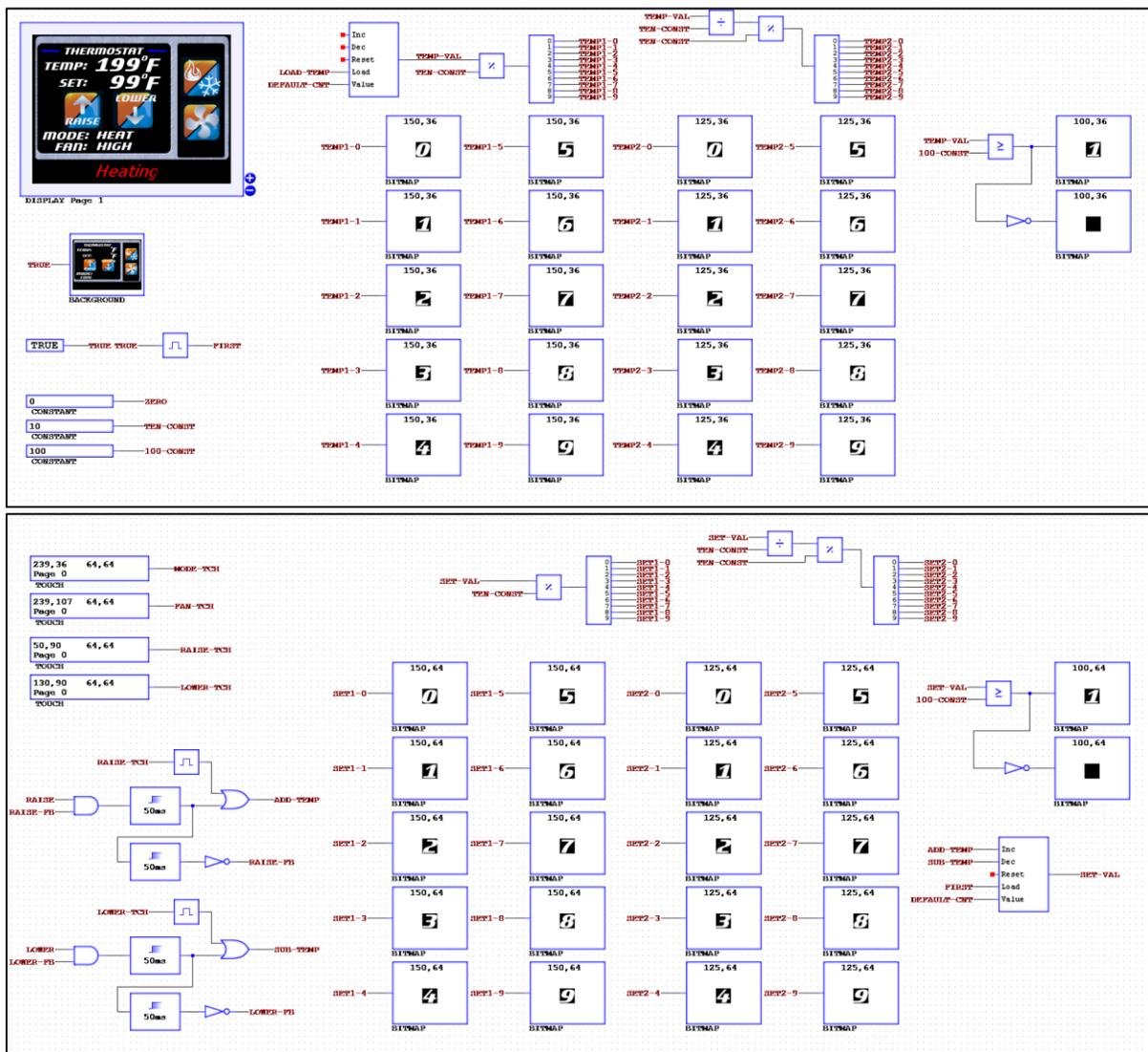
32. Finally, the bits have to be packaged into bytes to be transmitted. Insert 2 TOBYTE parts. For the first one, set nodes 0 to 3 as "Fan-off", "Fan-Low", "Fan-Med", and "Fan-High". All the rest of the nodes should be labeled as "False". The output should be "Led-Fan". For the second one, set nodes 0 to 2 as "Mode-Off", "Mode-Cool", and "Mode-Heat". Label the rest of the nodes as "False". The output should be "LED-mode". (If using a switch to test values and functionality of the program on hardware, these will tell the hardware what color to make the LEDs and what the mode/fan settings are.)

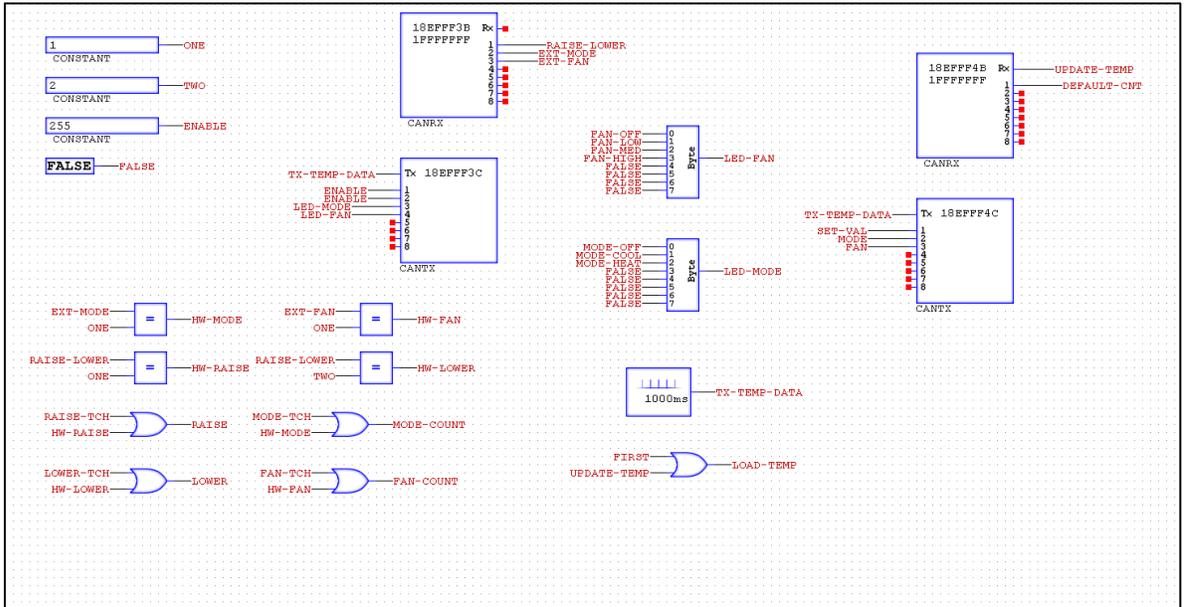
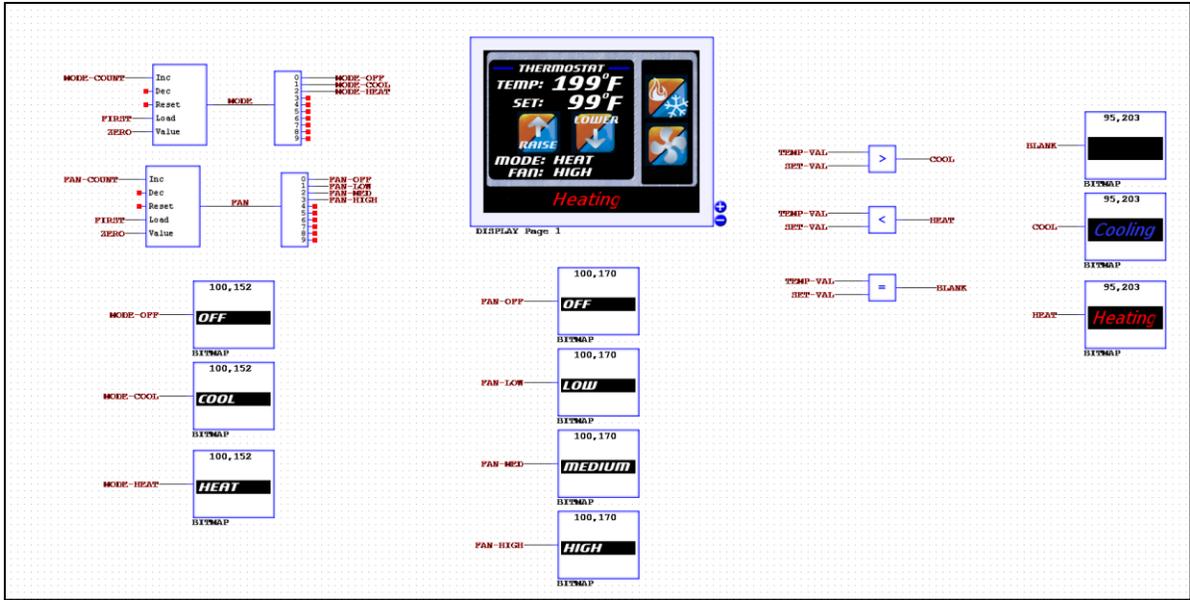


33. Run the simulator. The Default set and temp should be 0. Click Raise and Lower to change the set amounts. They can also be held down to change value without clicking. The Fire/Ice button will change the mode and the Fan button will change the fan speed. If the temperature is greater than the set value, “heating” will appear. If it is lower, “cooling” will appear.

**Note:** In order to change the temperature value in simulation mode, you must change the value on the CANRX object. Change “Default-CNT” to the value you want (ex:74) and then set “Update-Temp” to true, then to false.

## 9.3 Result



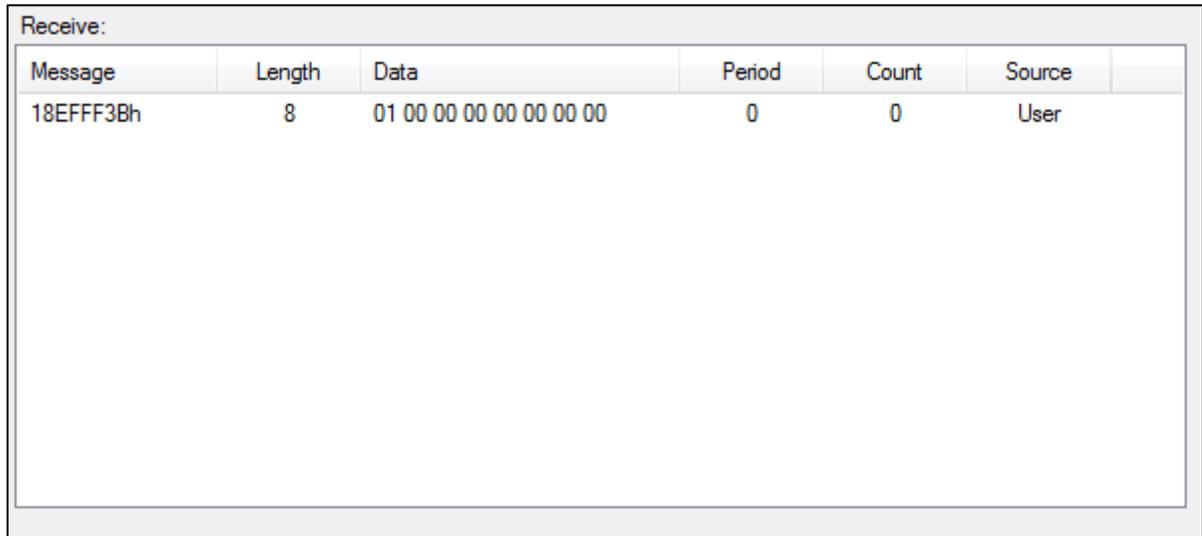


# Chapter 10 – simulating with Hardware

## 10.1 Using the Message Window

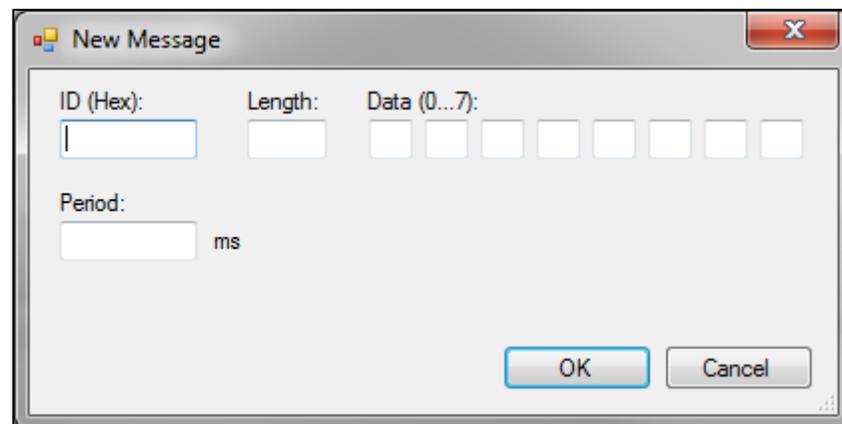
---

1. Locate the CANcreate file “chap\_9.ccs” or use your own created file from chapter 9. (Located under /Manual Tutorials)
2. Go to Connect and Show Message Window.



Message	Length	Data	Period	Count	Source
18EFFF3Bh	8	01 00 00 00 00 00 00 00	0	0	User

3. Under the Receive or Transmit boxes, right-click and options are shown. To create a new Receive message, right-click in the box and select new. A New Message window will pop up. There the ID, length, and Data can be inputted.



New Message

ID (Hex):

Length:

Data (0...7):

Period:  ms

OK Cancel

4. For this program, set the ID to “18EFFF3B”, length to 8, and data to “01 00 00 00 00 00 00 00”. Click OK. Observe that the source is User, but this will test as if the message was coming from a hardware device.

Receive:						
Message	Length	Data	Period	Count	Source	
18EFF3Bh	8	01 00 00 00 00 00 00 00	0	0	User	

5. Test with the same ID but with the following data. To change the mode, use data “00 01 00 00 00 00 00 00”. To change the fan, use “00 00 01 00 00 00 00 00”.
6. Just run the simulation and values will be received into the program logic.

## 10.2 Using a Switch

---

7. If you want to test using a switch to control variables without a touch screen device, use the following procedure.
8. Make sure the “chap\_9.ccs” is open, go to connect options and change the connection type to “514UTC USB”.
9. Set up the switch with preprogramed logic or navigate to /Switch Demo File and use the “usermanual-external-sw-cfg.crm” file to program the switches. Once all the addresses are set up, connect it to the computer via USB.
10. Enable CAN Simulation and turn the Simulator on. The switch should now control the increasing/decreasing of the temperature, changing the mode and the fan. The LEDs should change according to the settings of the mode and fan.
11. Open the Message Window to see the receiving and transmitting messages between the program and the hardware.
12. Both Mode and Fan are OFF:

Messages						
Receive:						
Message	Length	Data	Period	Count	Source	
18EFFF3Bh	8	00 00 00 00 00 00 00 00	1088	614	Hardware	

Transmit:						
Message	Length	Data	Period	Count	Source	
18EFFF3Ch	8	FF FF 01 01 00 00 00 00	1074	598	Design	
18EFFF4Ch	8	4A 00 00 00 00 00 00 00	1072	598	Design	

### 13. Change in Mode

Messages						
Receive:						
Message	Length	Data	Period	Count	Source	
18EFFF3Bh	8	00 01 00 00 00 00 00 00	112	846	Hardware	

Transmit:						
Message	Length	Data	Period	Count	Source	
18EFFF3Ch	8	FF FF 01 01 00 00 00 00	1007	810	Design	
18EFFF4Ch	8	4A 00 00 00 00 00 00 00	1007	810	Design	

## 14. Change in Fan

Messages						
Receive:						
Message	Length	Data	Period	Count	Source	
18EFFF3Bh	8	00 00 01 00 00 00 00 00	1025	899	Hardware	

Transmit:						
Message	Length	Data	Period	Count	Source	
18EFFF3Ch	8	FF FF 02 02 00 00 00 00	929	863	Design	
18EFFF4Ch	8	4A 01 01 00 00 00 00 00	929	863	Design	

## 15. Changes on Hardware



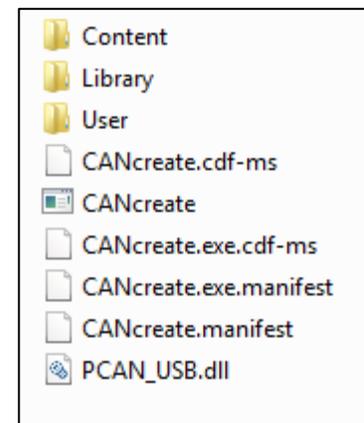
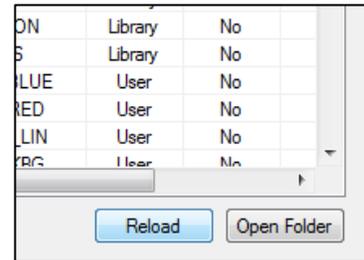
# Chapter 11 – Troubleshooting FAQ

## 11.1 Locating Installation Location

**Problem:** The CANcreate software just installed but you can't locate the execution folder that contains the application and Image Library.

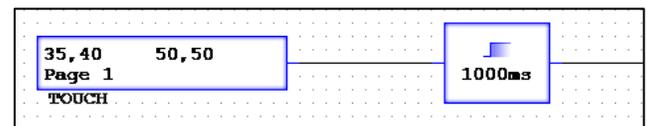


**Solution (CANcreate Version 2.0.0.58 or Below):** With Windows 7, run the CANcreate software. If it is not open, find the original extracted folder (ex: User\Name\Desktop\CANcreate if on desktop). Open the folder and double-click the "CANcreate" application run file. The software should open up. Next, go to tools and click "Open Image Library". Find the "Open Folder" button and click it. (Can also use the Open Application Folder button under Help->About) The folder that opens up is the application folder. You can choose to leave the folder in this location or copy all the contents into a newly created folder somewhere more preferable.



## 11.2 Preventing Accidental Touch

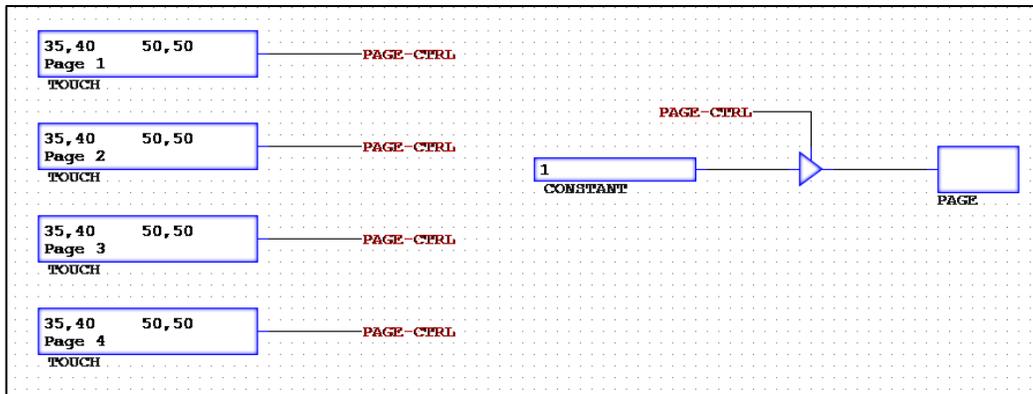
**Problem:** You and/or the daily users of a created program are consistently hitting touchable areas on the device unintentionally.



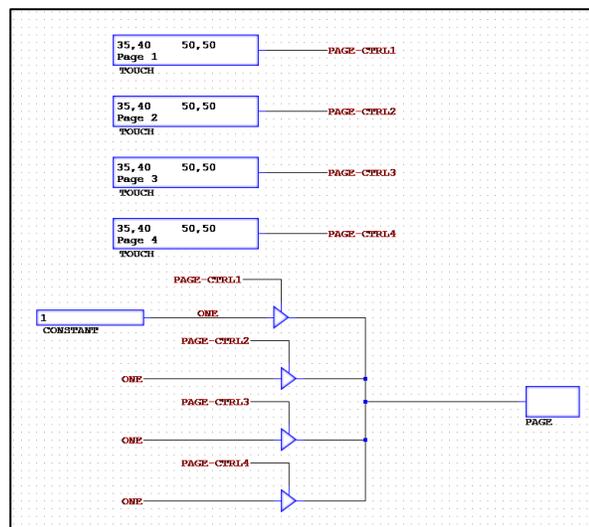
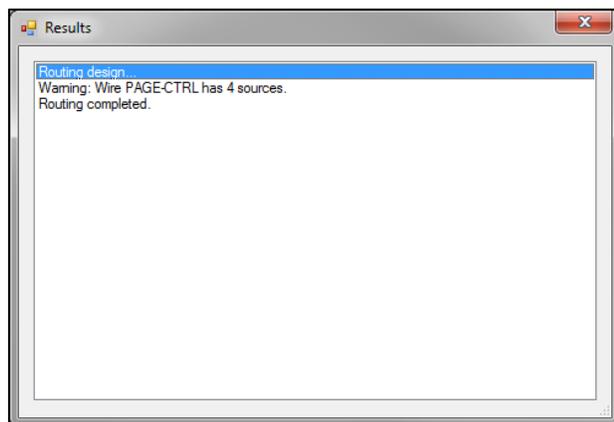
**Solution:** Open up the CANcreate file and locate the TOUCH parts that are giving trouble. Add a DELAYON right after the TOUCH part and before whatever function it is executing. Set the DELAYON timer to 100ms or higher (ex: 1000ms is a 1 second hold). This will then require the touch zone to be held for the specified period of time before going TRUE.

## 11.3 Touch Parts “Not Working”

**Problem:** When running the program in the simulator or on device, the TOUCH areas won't respond to input. They seem not to work and there is no visible sign of the TOUCH part going TRUE while in the CANcreate application.



**Solution:** This problem is most likely due to multiple sources (TOUCH parts) to one wire and its function. To check this, route the program, and then go to “Route Results” under Publish. There should be a warning where it states what wire has the X number of sources. **This can occur even if the TOUCH parts are assigned to different pages, they are still all connected to the same wire.** There are two methods of fixing this problem. One way is to set different DELAYON timers for each TOUCH part in your program. The second way is to give each TOUCH part an individual wire name and duplicate the same code for each to use. Re-route and make sure the warning has been fixed.

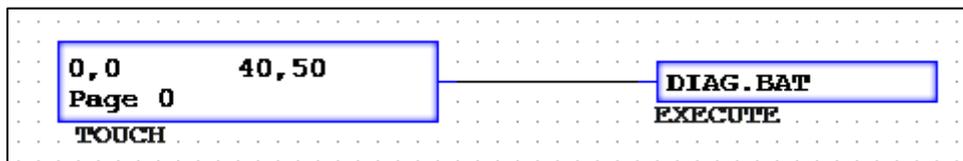


## 11.4 Device Time is Incorrect

---

**Problem:** When running the program on the device, the clock and/or calendar is incorrect. Yet, the clock and calendar were correct in the simulator.

**Solution:** This occurs because the simulator uses the computer's system clock and calendar when running the program. On the device, there is no built in clock and calendar, so it must be set using the "Diag.bat" described in Chapter 1. Your program must contain a TOUCH part attached to an EXECUTE part with file "Diag.bat". This will enable the setting of the time and date when the program is run from the device.

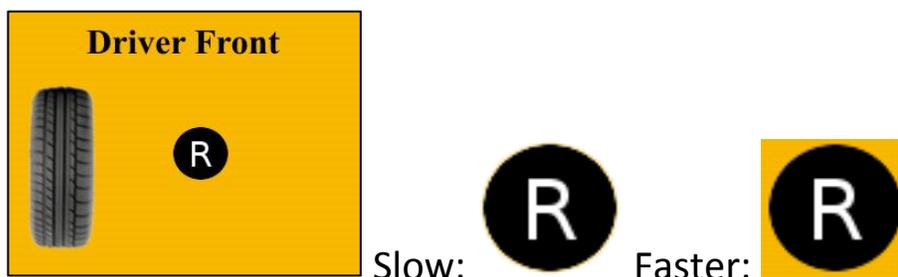


## 11.5 Bitmap Rendering Too Slow

---

**Problem:** The Bitmap is rendering onto the device's screen very slowly.

**Solution:** This can occur from a couple of factors. One, the bitmap can be too large of an image, or, two, the bitmap has transparent properties. For the first factor, this can be solved by chopping the bitmap up into smaller bitmaps. The smaller images will load faster and speed up image rendering. For the second factor, the bitmap that under part properties has the box for transparency checked can be changed to load faster. **Using a bitmap with transparency requires more cycles within the program to draw the image because it needs to duplicate the background in the transparent sections.** To fix this, one can edit the image and replace the transparent sections with the correct background that would be behind it. This will speed up rendering.

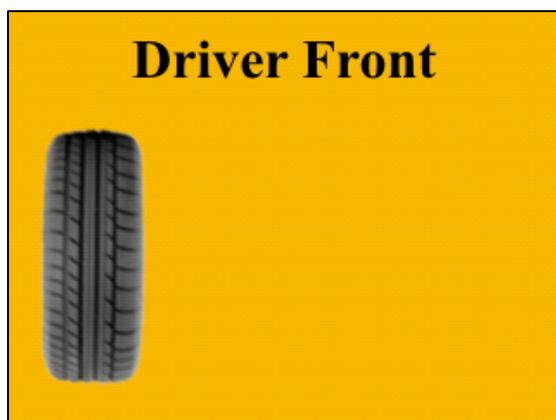


## 11.6 Bitmap Won't Disappear

---

**Problem:** The Bitmap is set to true then false but won't disappear from the screen.

**Solution:** This can occur in some situations, mainly in terms of trying to cover up information on the screen and remove bitmaps from view. To fix this, the simplest solution is overwriting the bitmap with another bitmap. So create a bitmap from the background or something that makes sense, set it to the same X and Y, and set it to TRUE.



Cover with



## 11.7 Why Bitmaps are Renamed

---

**Problem:** When looking in the folder of a published program, the images are all renamed to shorter names.

**Solution:** This happens because this is part of the publishing process. The script requires the image names to be 8 characters or less. It is best to get accustomed to naming all images beforehand in this form.

## 11.8 Bitmap Image Support

---

**Problem:** Bitmap image is not showing up when running on device.

**Solution:** The only supported image format is PNG images. Other image types aren't accepted and won't be displayed to the screen correctly.

## 11.9 Tutorial Images not Displayed

---

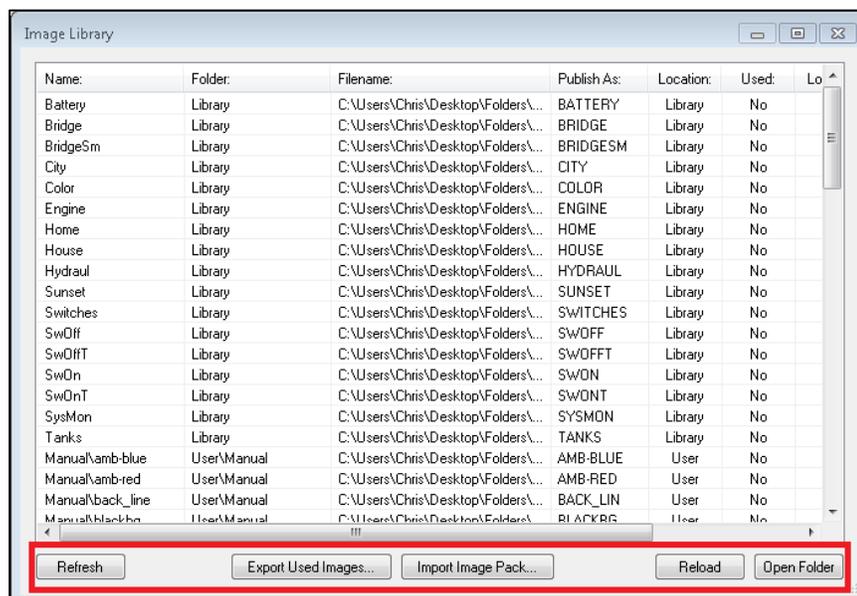
**Problem:** Background and Bitmap images don't show up when a premade tutorial is opened. All are replaced with red 'X' images.

**Solution:** Make sure that the "Manual Images" folder is in the User folder where the CANcreate software runs from. Once folder is copied over, the images should appear in the tutorial correctly.

## 11.10 New Library Images not Listed

---

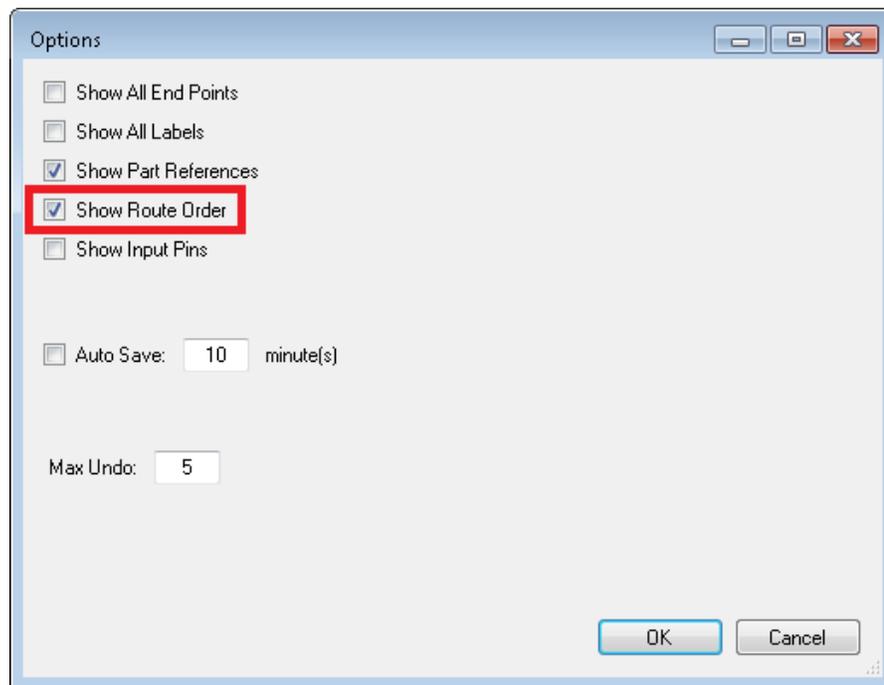
**Problem:** After putting new images into the User folder, they don't appear in the Image Library when placing a Background or Bitmap.



**Solution:** Open the Image Library (see above), click Refresh and then Reload to have CANcreate software import the newly added images.

## 11.11 Design not working Correctly

**Problem:** The program is not behaving correctly or executing logic as it should be.



**Solution:** Make sure the design is routed and use the Tools Options, then check show Route Order. Verify that it has a value that isn't "-1". It possible to simulate some designs without routing but in order to execute logic, it needs to be routed.

