

# STEMSEL Beginners Project 1: Alarm System

## Problem

How can we use microchips to protect our valuables? How can a microchip know if there is a thief and what can it do when it detects one?

## Background

Have you heard of a microchip before? Do you know what microchip means? Micro means very small, and a chip is a small piece, like a potato chip or a wood chip. A microchip is actually a very small piece of silicon, and people can use that tiny silicon computer to store a list of instructions to make the microchip do many different things. Microchips are all around, in everything from computers to cars, credit cards to toys. One thing that microchips are very useful for is alarm systems, since unlike a person the microchip can't get bored of watching the same thing for hours on end.

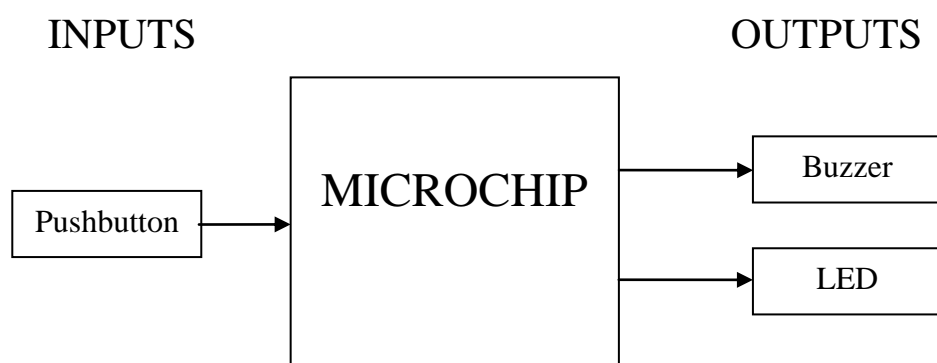
Alarm Systems are used to protect a wide variety of valuables, from a car to household belongings. Alarms use sensors which can be a simple switch or more complex heat, light and motion sensors. A key component of an alarm system is a microchip to gather sensor information and decide when to activate the alarm.

## Ideas

Look at the STEMSEL controller board. Can you see any inputs, i.e. something that we can touch or change to tell the microchip something? What about an output, i.e. something the microchip can change to tell us something? What kind of inputs and outputs are normally on an alarm system? What inputs and outputs can we use on our alarm system?

## Plan

For our alarm system we can use the push button as our input, which will tell the alarm to turn on. We can use a buzzer and the on-board LED to show that the alarm has been activated.



# **Part A: Design and Build the Alarm System**

## **Step 1: Design the project**

Start ezCircuit Designer from the desktop.

- a. On the menu click “File” and select “New”.
- b. In the Name field type “AlarmSystem” and click “OK”

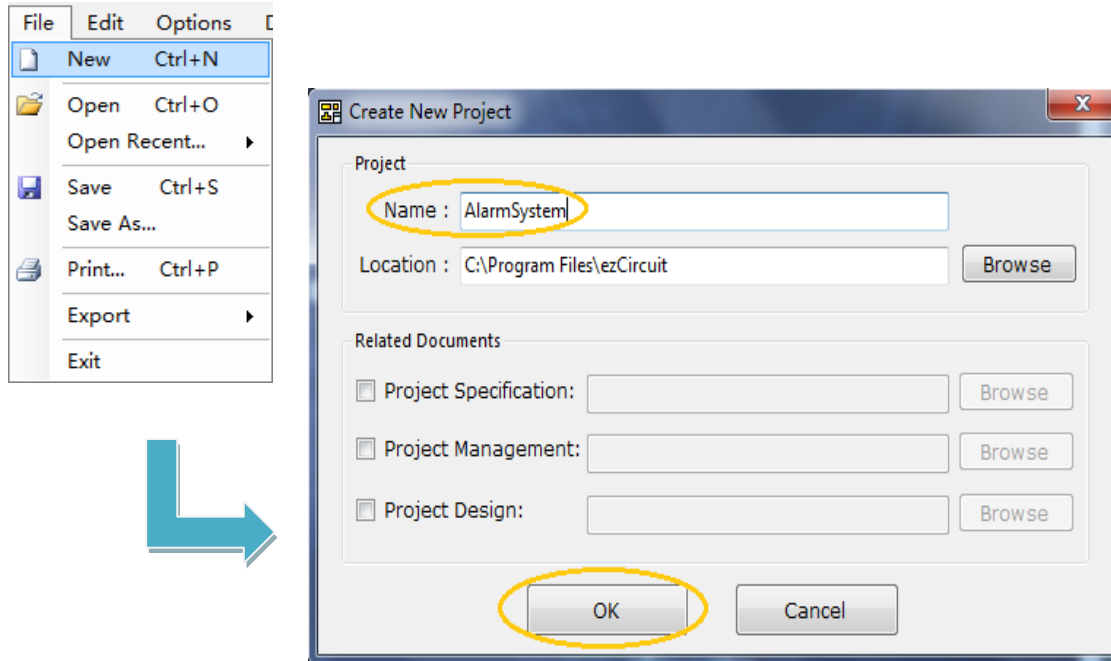


Figure 1.1

- c. If the STEMSEL controller board is already selected go to step e. Otherwise click on the “Options” drop down menu click “Chip Selection...”.
- d. In the Chip Selection window click on “STEMSEL” in the Model column and then click “OK”. When a confirmation window appears click “OK”

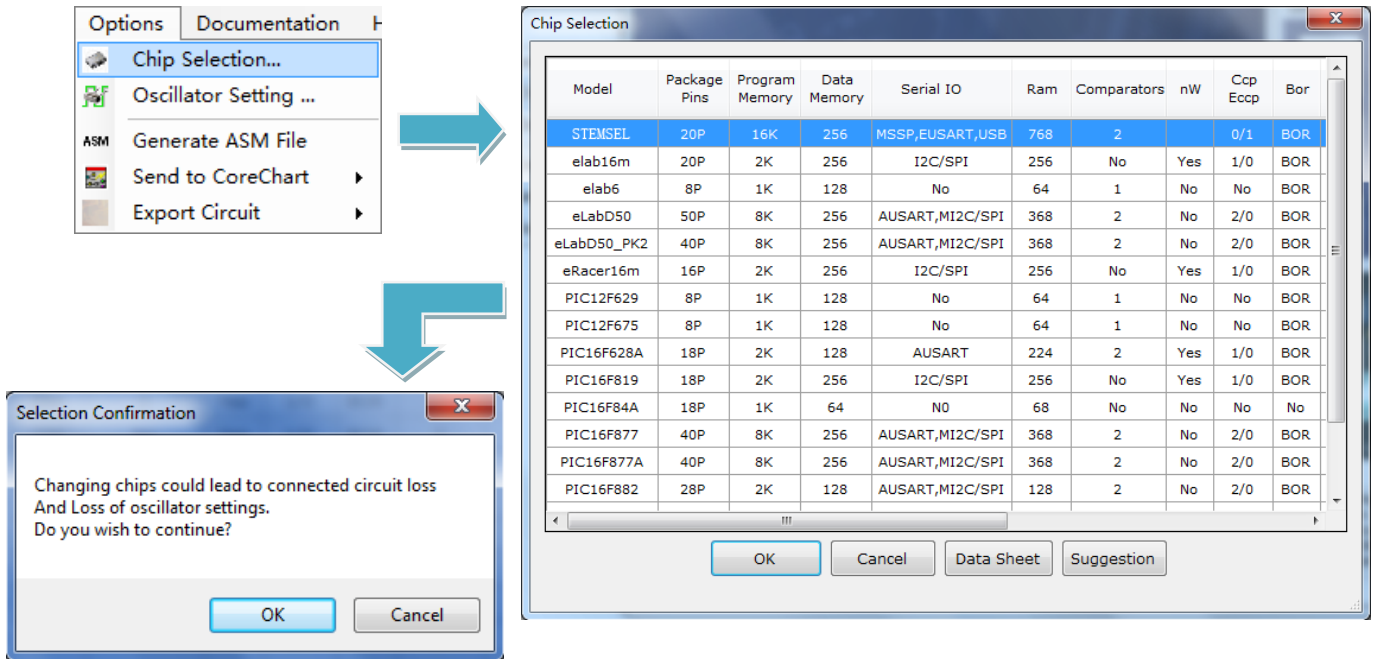


Figure 1.2

A symbolic representation of the STEMSEL controller board is then displayed on the screen. This makes it easier to connect circuits from ezCircuit Designer to the STEMSEL controller board.

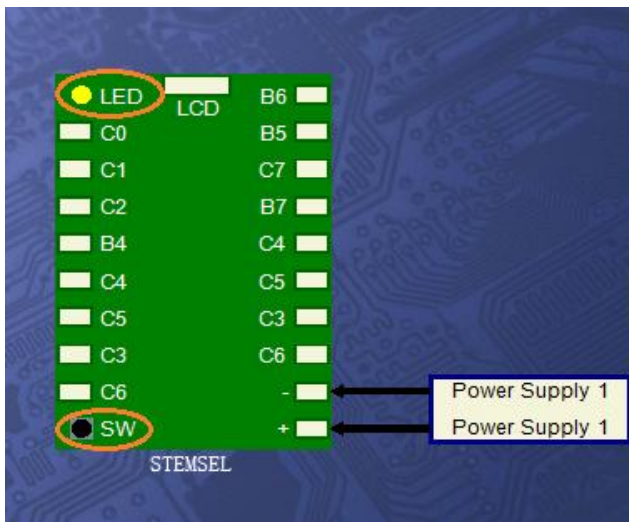


Figure 1.3

Note - Connections B6, B5, C7, B7, C4, C5, C3, C6, “+” and “-” are located on the green connector strip while the others are available on the Black pin connector.

The Yellow LED and the push Button switch are built on to the printed board.

- e. Click on the Buzzer icon, the name will then appear, highlighted in the Circuits window. Then click on the pin you want to attach it to, in this case “C4”.

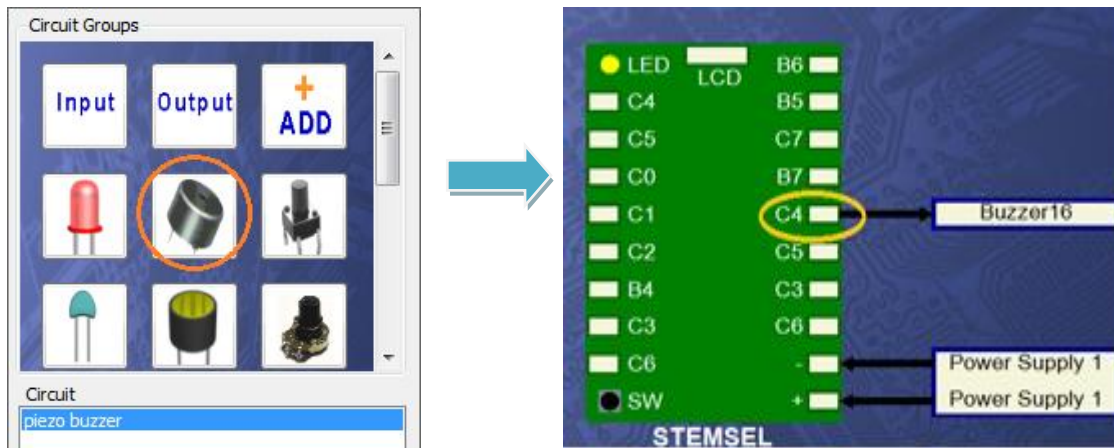


Figure 1.4

- f. Rename the buzzer to just “Buzzer” by right clicking on the “Buzzer16” label and selecting “Rename”, typing the new name then pressing enter on the keyboard.

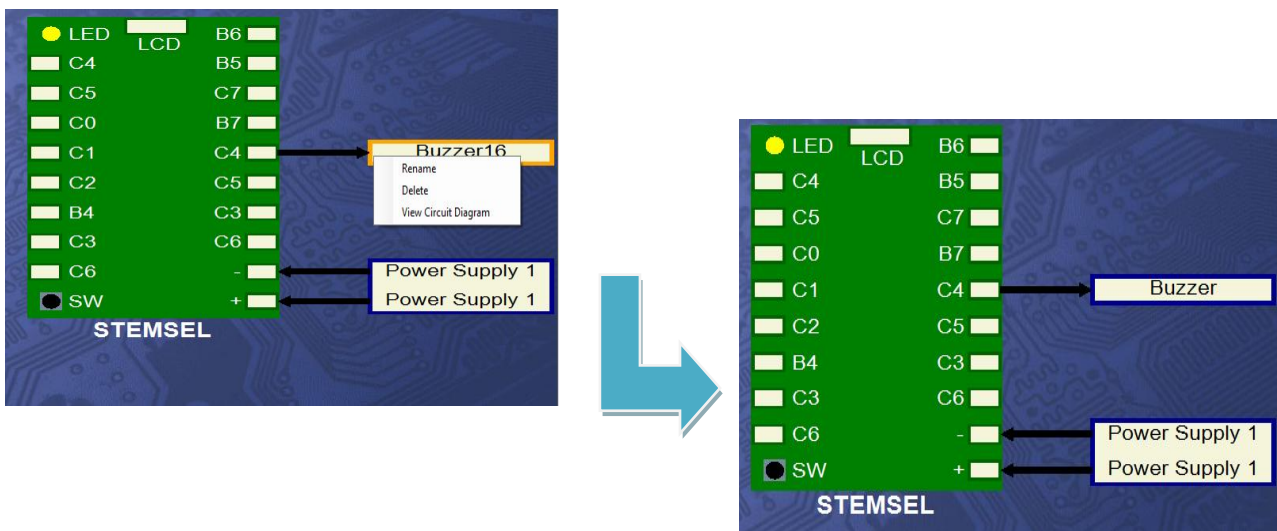


Figure 1.5

The yellow LED and the push button are part of the STEMSEL controller board and do not need to be added. This completes the ezCircuit Design stage.

## Step 2: “Build” the project

The buzzer and power supply (batteries) now need to be connected to the STEMSEL controller board.

- a. Place the three AA batteries into the battery holder and connect the power pack to the white socket (circled in figure 2.1), next to the ON/OFF switch on the STEMSEL controller board.

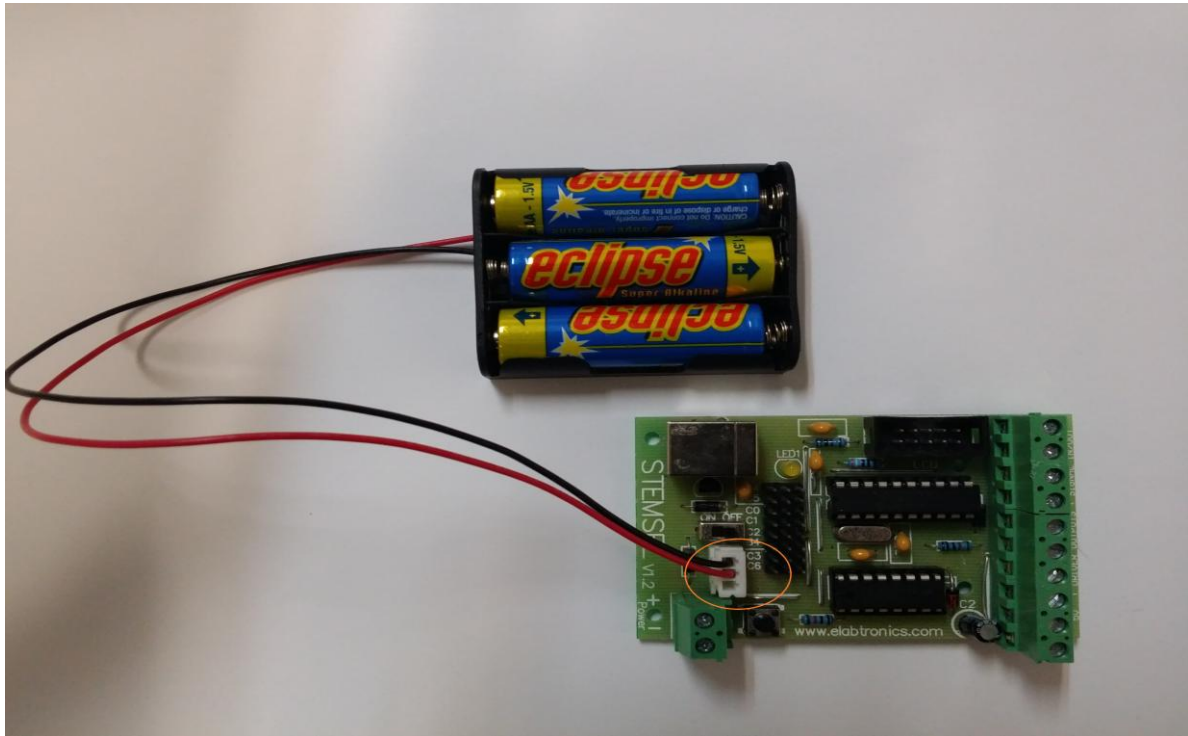


Figure 2.1

- b. Connect the black wire of the buzzer to the “-“ (negative terminal) on the green connector strip and connect the white wire to terminal “C4” (see Figure 2.2). Remember to turn the screw anti-clockwise to open the terminal, and clockwise to close it.

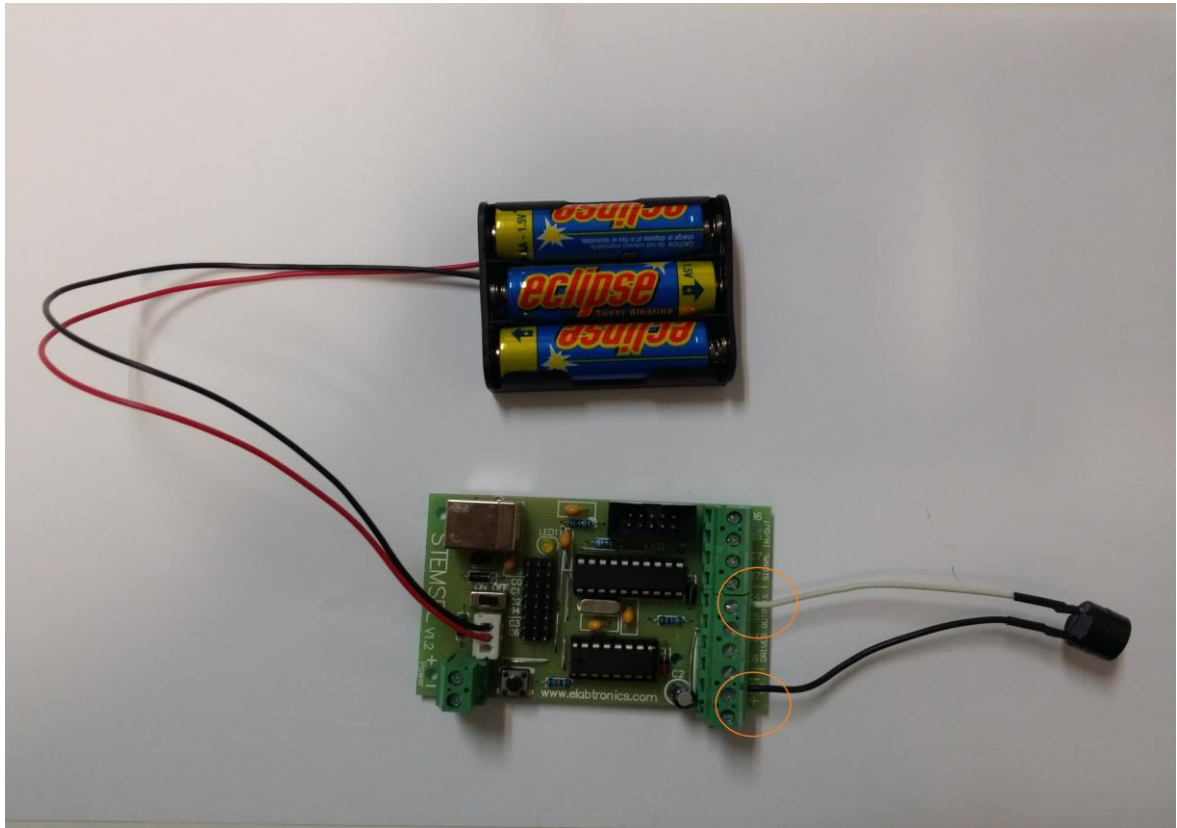


Figure 2.2

- c. Connect the programming cable to your STEMSEL controller board as shown on Figure 2.3 below. Do not connect the other end for now.

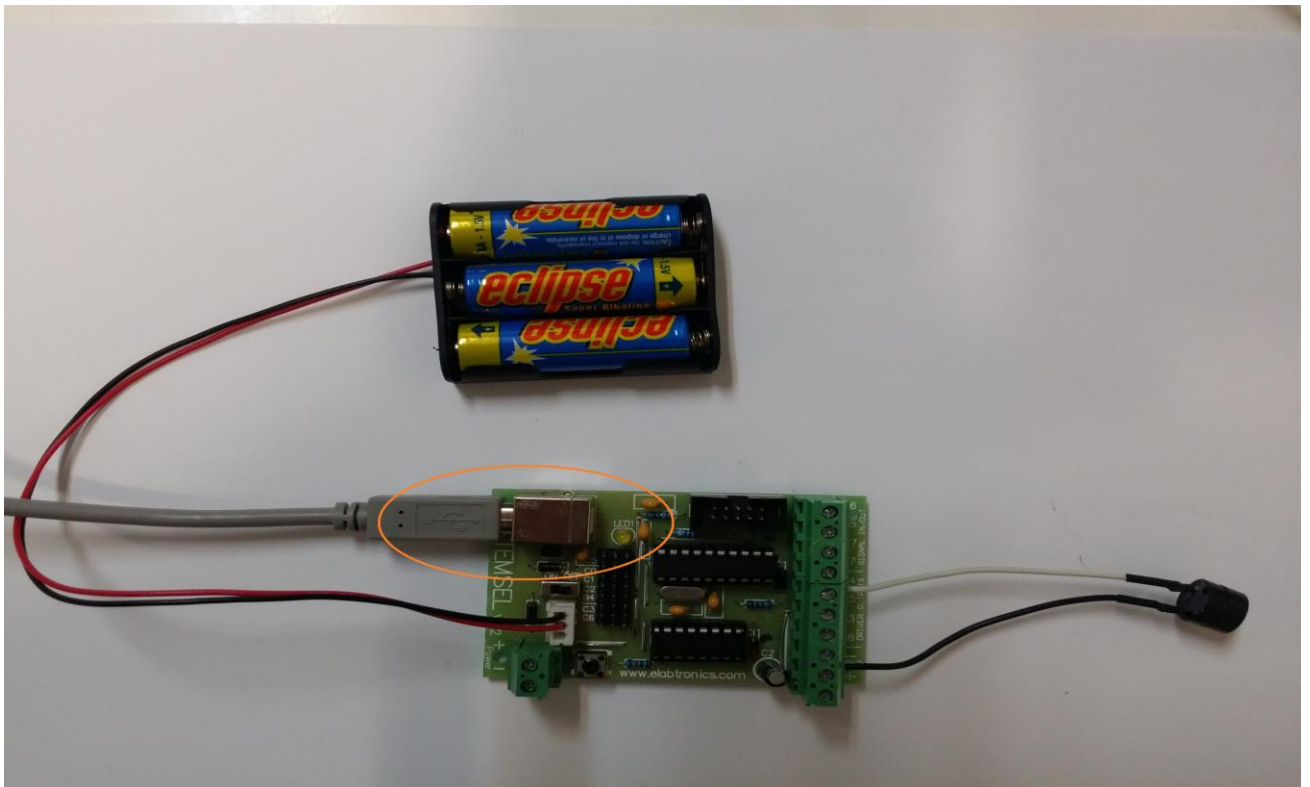


Figure2.3

## **Part B: Test and Program the Alarm System**

### **Step 3: Export design to CoreChart**

ezSystem uses ezCircuit Designer to connect components to CoreChart. Note: component names assigned in ezCircuit Designer are used in CoreChart to program the microchip.

- a. Ensure ezCircuit Designer is opened. If not, open the “Alarm System” project by clicking on the “File” drop down menu & select “Open”. Navigate to the directory where the project was saved. Select the project & click the “Open” button.
- b. Click the “Send to CoreChart” icon in the icon bar.

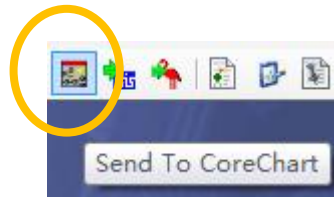


Figure 3.1

Before the microchip can be programmed a test needs to be done to ensure the buzzer is correctly connected to the STEMSEL controller board and is working correctly.

The test program looks like Figure 3.2. When a design is sent from ezCircuit Designer to CoreChart, a test subroutine is automatically generated for each of the output circuits in the design.

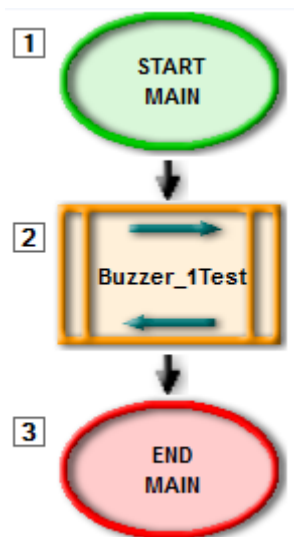


Figure 3.2

## **Step 4: Test the outputs**

A test subroutine is automatically generated to check that the output buzzer circuit is working properly.

- a. Save the CoreChart program. On the “File” drop down menu click “Save As...”. Type the program name “Alarm\_System” and click “Save”.
- b. Check that the battery pack is connected and the power switch on the STEMSEL controller board is set to “OFF”. While holding down the black button on the board, connect the USB end of the programming cable to a USB port of your computer. See the Beginner Project Notes for more information about sending programs to the STEMSEL controller board.
- c. Click the “Send Program To Chip” button. When prompted to save the program click “OK”.

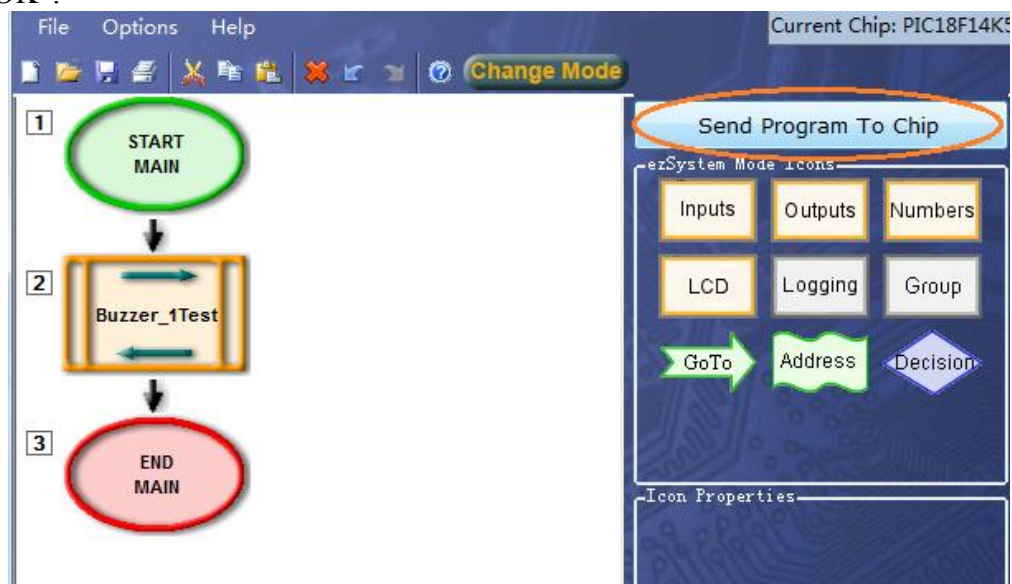


Figure 4.1

- d. After the assembly code is generated by CoreChart the program will download to the chip. Once the download has completed a “Programming was successful” message will display. If the “Device not detected” message is displayed, return to Step b. and ensure you press and hold the button on the STEMSEL controller board while connecting it to computer.
- e. Disconnect the STEMSEL controller board and after a few seconds switch the power to “ON”
- f. The test program should now automatically run.

This test subroutine will turn the buzzer on briefly and then turn the buzzer off. If the buzzer is not turned on and then off, turn the STEMSEL controller board off then on again. If this does not work then:

- Check the connections to the green connector. (see Beginner Project Notes for connecting wires into the terminals)
- Check the battery pack is plugged in
- Reconnect the STEMSEL controller board to the PC and press “Send Program To Chip” again



## Step 5: Program the STEMSEL Controller Board

The hardware testing is complete. We will proceed to program the microchip with CoreChart which enables users to quickly create programs for microchips.

The alarm program will turn the buzzer and yellow LED on for five seconds and then turn them both off.

- a. Delete the buzzer test subroutine by clicking on the icon and selecting “Delete” in the icon bar. Alternatively, use the Delete key on the keyboard.
- b. When prompted to confirm the delete click “OK”

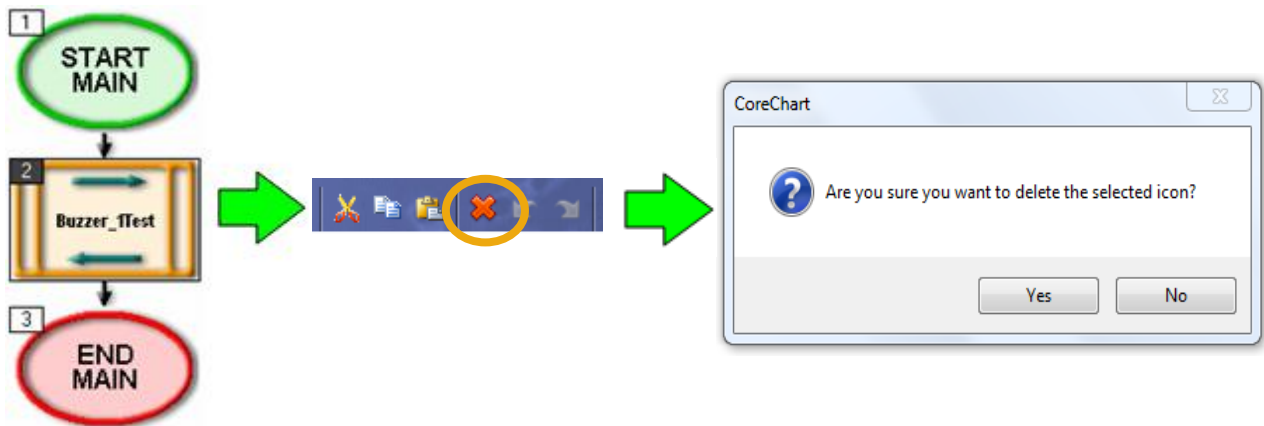


Figure 5.1

- c. Select the “Outputs” button in the “ICONS” menu and click the “OnOffPin” item in the “ICON Properties” list.
- d. Click on the black arrow between the “START MAIN” and “END MAIN” icons to place the icon.

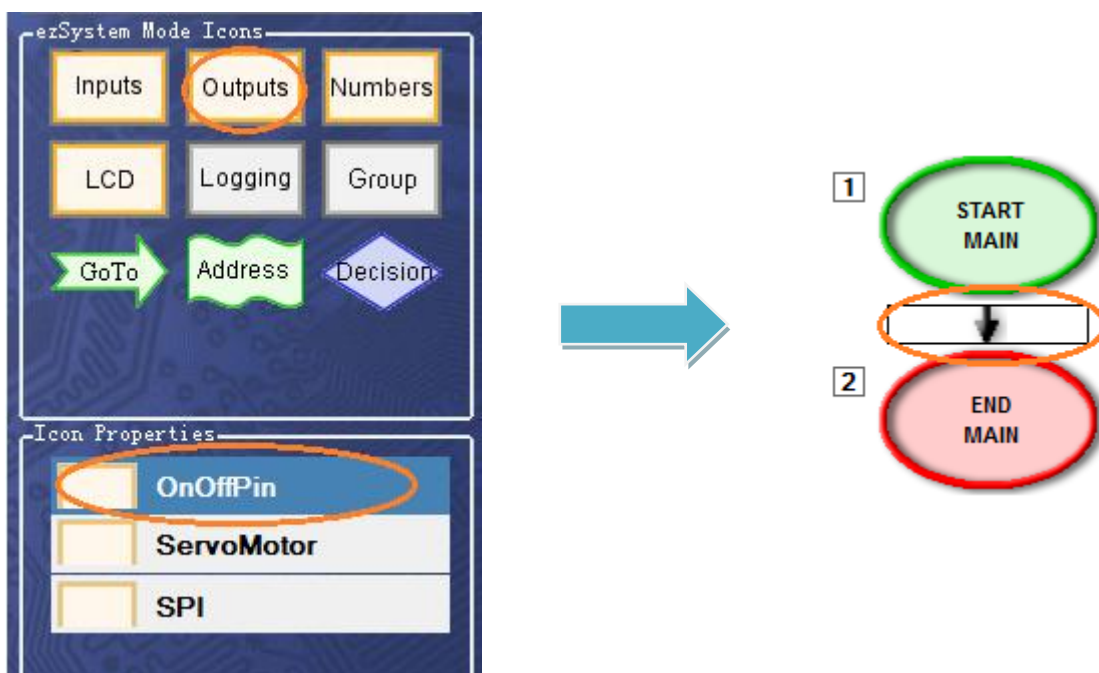


Figure 5.2

- e. Double click on the “OnOffPin” icon that was placed to open a form with the icon options.
- f. In the “OUTPUT PIN” field select “Buzzer” and select “ON” to turn the buzzer on. Click “OK”.

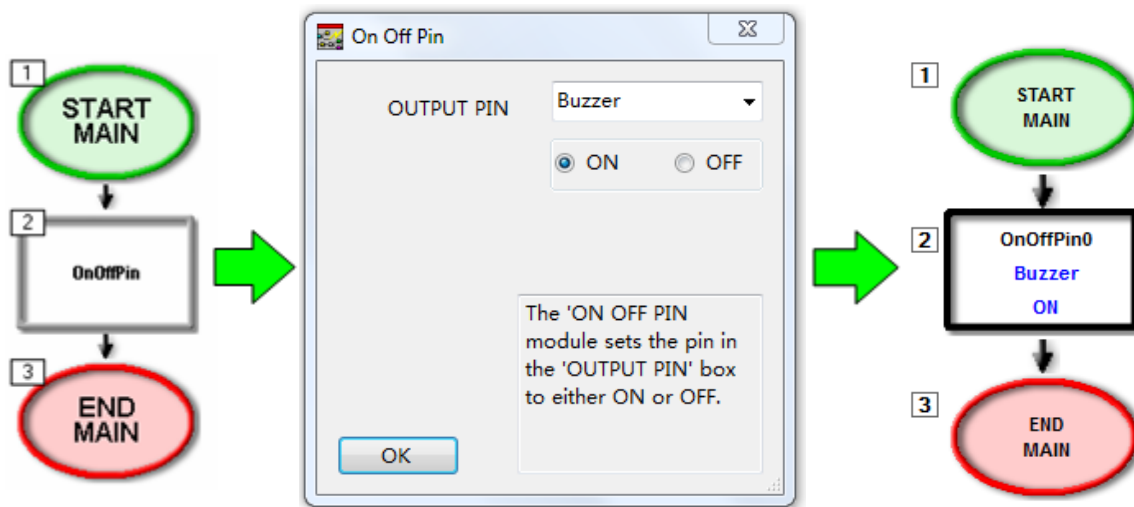


Figure 5.3

- g. Click “Outputs” in the “ICONS” menu and select “OnOffPin” in the “ICON Properties” list.
- h. Place a second “OnOffPin” icon below icon 2 by clicking the black arrow below the second icon.
- i. Double click on the new grey icon and select “C0\_YellowLED” in the “OUTPUT PIN” field. Ensure that “ON” is selected before clicking “OK”.
- j. Click the “Numbers” button in the “ICONS” menu and select “TimeDelay” from the “ICON Properties” list.

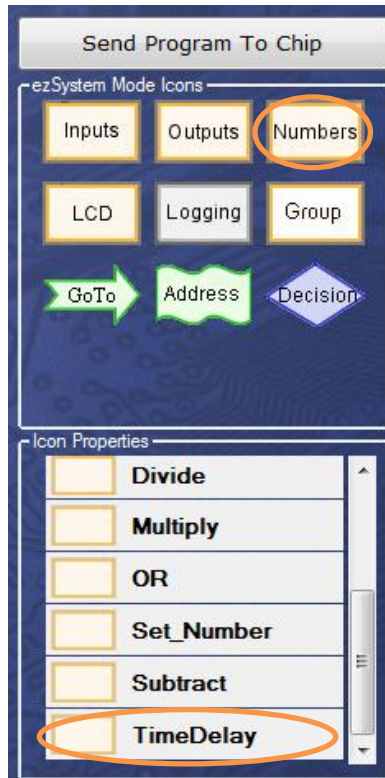


Figure 5.4

- k. Place the “TimeDelay” icon after the first two “OnOffPin” icons by clicking the black arrow below icon 3.
- l. Double click on the “TimeDelay” to open the options form. Type the number 5 into the “SECONDS” field.

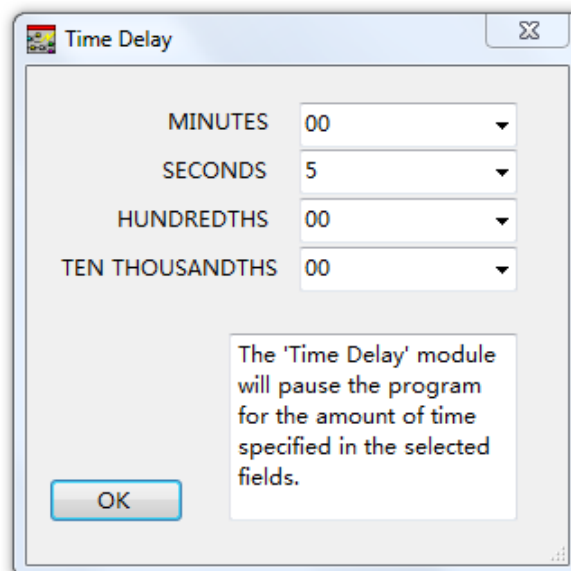


Figure 5.5

- m. In the “ICONS” menu select another “OnOffPin” icon from the “Outputs” group. Place it at the end of the program, below the “TimeDelay” icon. Configure this “OnOffPin” icon to turn the “Buzzer” to “OFF”.
- n. Place another “OnOffPin” icon below icon 5 to turn the “C0\_YellowLED” to “OFF”. Your program should now look like this:

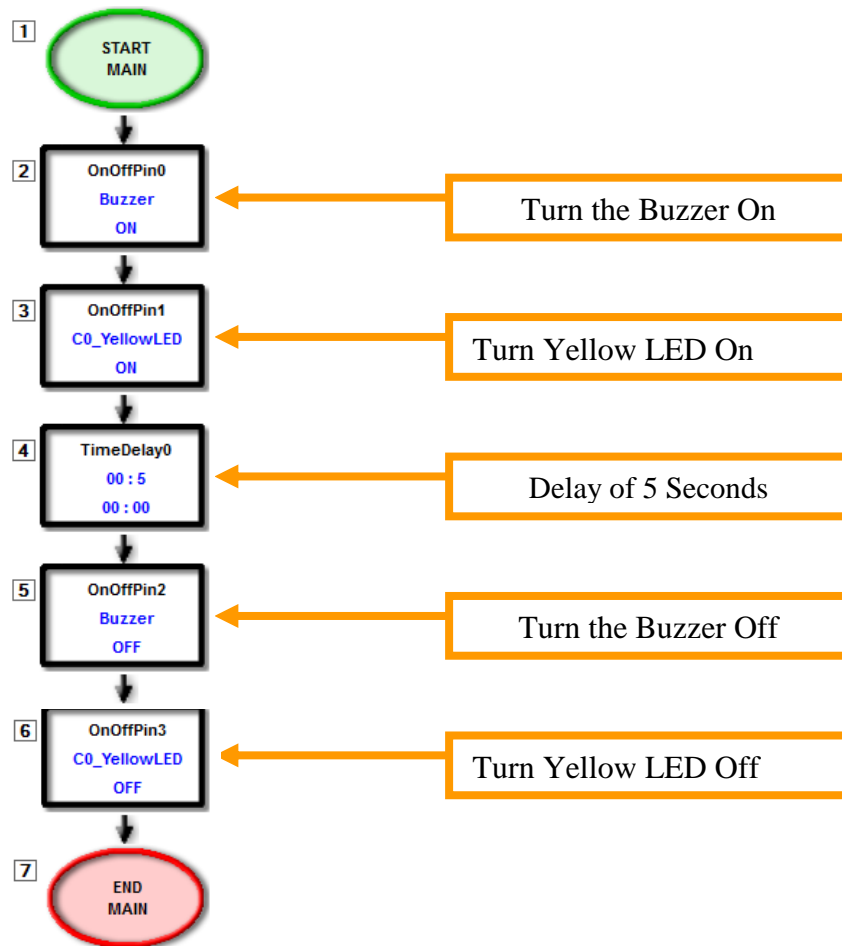


Figure 5.6

- o. Check that the USB cable is connected to the STEMSEL controller board, and that the USB end is connected to the USB port on the computer. Please remember to switch OFF the STEMSEL controller board, and to press and hold the push button while connecting.
- p. Also check that the battery pack is connected.
- q. Click the “Send Program To Chip” button and click “OK” when prompted to save the program.
- r. Wait for the program to download to the STEMSEL controller board.
- s. Disconnect the USB cable from the STEMSEL controller board. Wait a few seconds and turn the power ON.
- t. The program should now start running.

The program will turn the buzzer and yellow LED on, delay for five seconds and then turn them both off.

## Step 6: Incorporate Input Sensor to the Program

A switch is used as an input sensor for the alarm system. When the push button switch is pressed the alarm will sound. A decision icon is used to check if the switch is pushed.

- a. Click the “Decision” button in the “ICONS” menu.
- b. In “ICON Properties” select “A3\_PushButton” in the first box and select “OFF” in the second box. “OFF” in this case means that we want to know when we are pressing the Push Button. (See beginner notes for more information about the pushbutton).
- c. Place the decision at the beginning of the program, below the “START MAIN” icon, by clicking the black arrow below it.

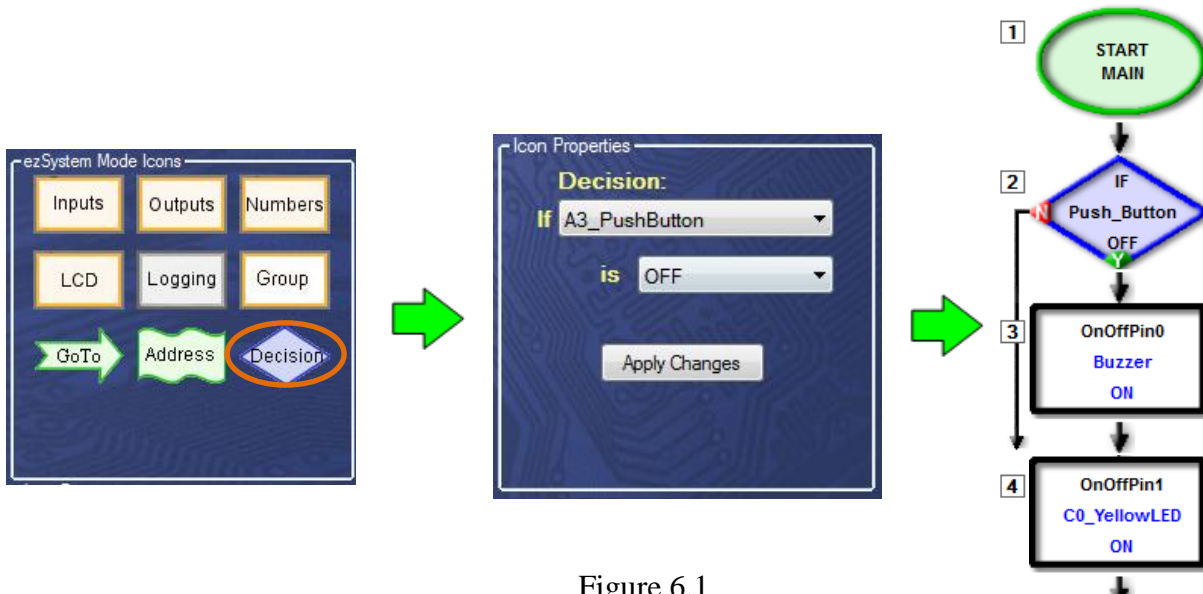
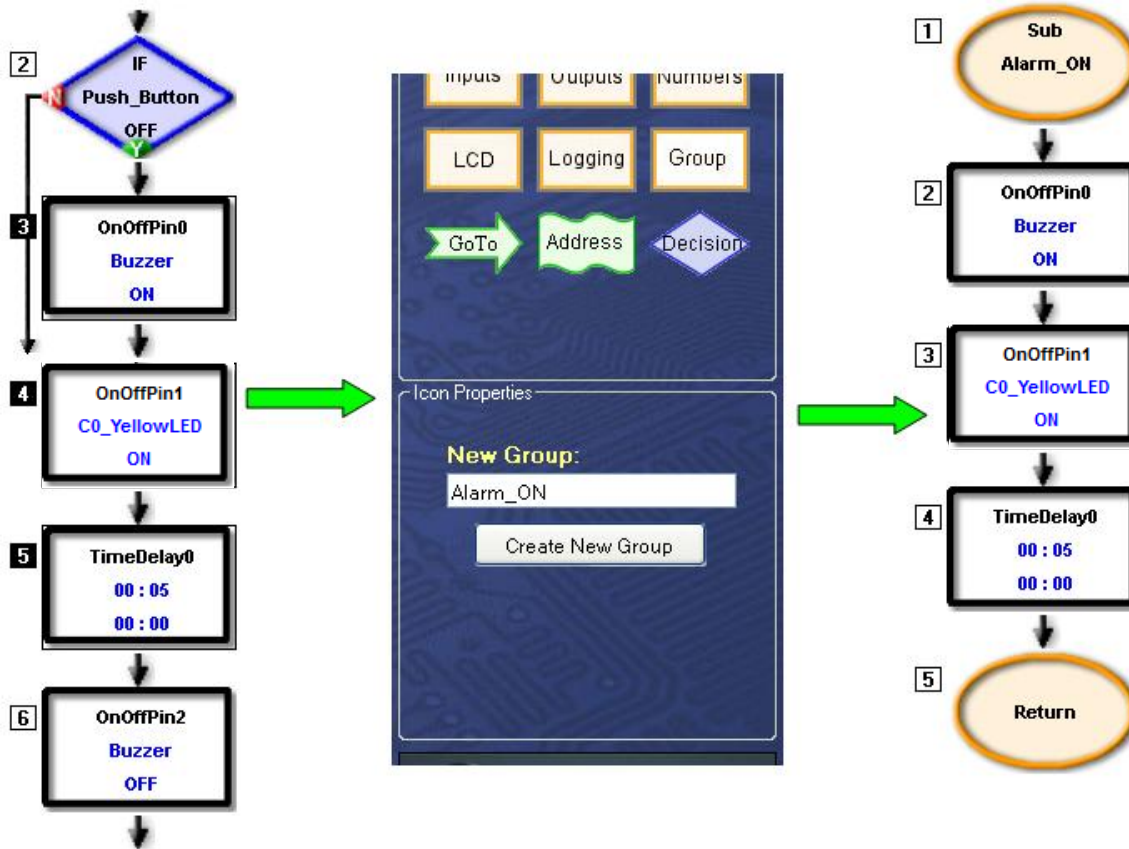


Figure 6.1

The decision icon was placed in the program in Fig 6.1 to detect whether the Push Button is pressed. The program will only now turn on the buzzer if the pushbutton is pressed. In order for the decision icon to turn on the buzzer, the LED and the time delay when the button is pressed, the group function can be used.

The GROUP function is used to group multiple icons into a subroutine or single icon. Follow these steps to use the group function:

- d. Click on icon 3 to select it, then while holding down the Shift key on the keyboard click on icon 5. This will select icons 3 through to 5. Then release the Shift key.
- e. Click on the “Group” button in the “ICONS” menu. Select “Group\_Name” and type “Alarm\_ON” as the name of this group. Click the “Create New Group” button.



You can view the contents of the new group by double clicking on it. To return to the main program, double click on either the first icon, “SUB Alarm\_ON” or the last icon, “Return”.

The alarm system program is still unsatisfactory because the program will only run once. When the program starts it will check the push button switch. If the push button switch is pressed then the buzzer and the LED will turn on. After five seconds they will both turn off and the program will end. On the other hand if the push button switch is not pressed the program will end immediately. This problem can be fixed by using a “GoTo” icon to create a loop to run the program continuously.

- f. Click the “GoTo” button in the “ICONS” menu. In the “ICON Properties” the box next to the label “GoTo” should have “START” selected.
- g. Place the icon at the end of the program by clicking the last black arrow.

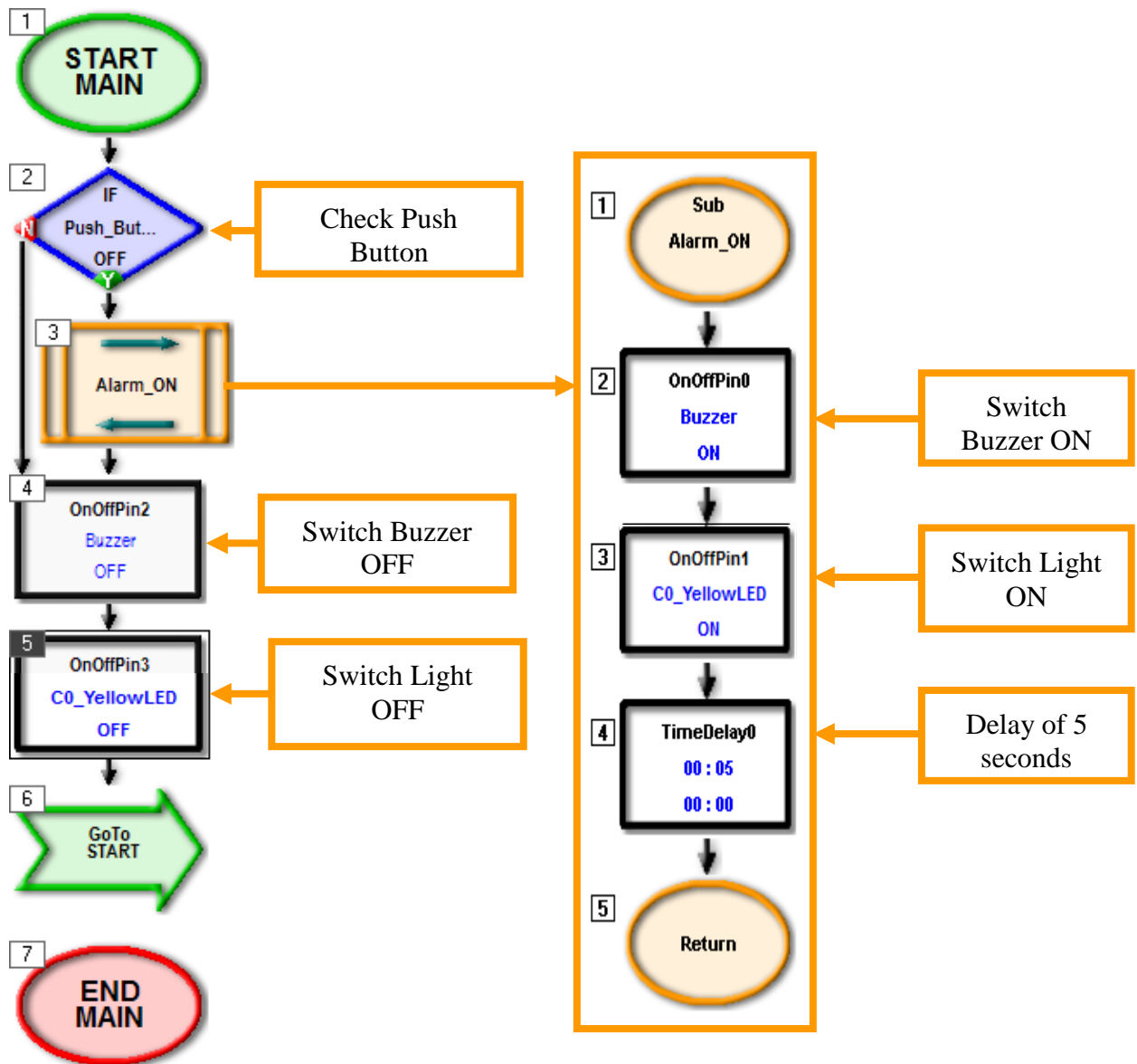


Figure 6.4

- h. Save the program, make sure the hardware is connected properly as described earlier in this workbook and send the program to the chip.

The program will turn on the buzzer and yellow LED for five seconds when the push button switch is pushed.

## **Part 7: Extend the Alarm System Program**

The alarm system now works as a basic alarm that will activate when the push button switch is pressed. There are still some areas where the alarm system can be improved. Work through the following challenges and devise a solution to the problem. Make small changes to the program and run the program regularly as you progress.

### **Extension 1**

As it is, the alarm will only sound while the push button switch is pressed i.e. the sensor is activated continuously. The alarm will stop when the switch is no longer pressed or released. This is not very satisfactory.

1. Modify the Alarm System so that once the alarm is activated it will continue to sound until it is reset by turning the power off and then back on. You can actually do this just by adding a GoTo icon inside the Alarm\_ON group.

The alarm system is still not satisfactory because if the alarm is activated while the owner is away on holidays, the alarm will sound continuously disturbing the neighbours. The alarm system may not be complying with noise pollution laws if it sounds for a long period of time continuously after it has been tripped.

2. Modify the Alarm System so that when the alarm is activated the buzzer and yellow LED will be on for five seconds and off for ten seconds and then continue to loop until reset. Again, you should only need to make changes inside the Alarm\_ON group.

### **Extension 2**

The alarm system has been improved. However new features and functions can still be added by changing the program in CoreChart.

3. Modify the Alarm System so that when the alarm is tripped the yellow LED will flash quickly. The buzzer will continue to be on for 5 seconds and off for 10 seconds.

**Hint:** Use a group of icons that will flash the yellow LED a number of times. This new group of icons can then be used to replace the 5 second time delay. Remember,  $10 \times 0.5 \text{ seconds} = 5 \text{ seconds}$ .

## **Summary**

Microchips are basically small simple computers that we can use to do many different things. We actually use microchips every day in countless devices, such as alarm systems. In this project we built an alarm system that used a pushbutton as an input, and would activate a buzzer and LED to let us know that the alarm had been triggered. This project should have given you a good background in how to program the STEMSEL controller board using ezSystem, which can be used together to make lots of other exciting and useful things.