

STEMSEL LCD Project 6 : Clock

Problem

Design a program that displays a clock on the LCD.

Background

Modern society already knows that it is surrounded by hundreds of microchips, doing all sorts of tasks such as waking people up in the morning, preparing food, playing games, or checking emails. However, sometimes humans do not appreciate the programming behind even the most simple-looking devices.

Clocks are something we often take for granted, but have you ever thought about how it works? Our clock will need to be able to not just simply keep and display the time correctly, but will also need some sort of input so that we can set the correct time.

Ideas

What information does a clock need to display? When does the clock need to increment the numbers? How can users set the time? How is it possible to check if the clock is working properly; do we need to keep the microchip running for a full day?

Plan

It is possible to a LCD to display the hour and the minute. To be able to set the time effectively, 2 buttons are needed. When running, the “hour” variable will increment once “minute” reaches 60 and both “hour” and “minute” will need to be reset once they reach 13 and 60 respectively.

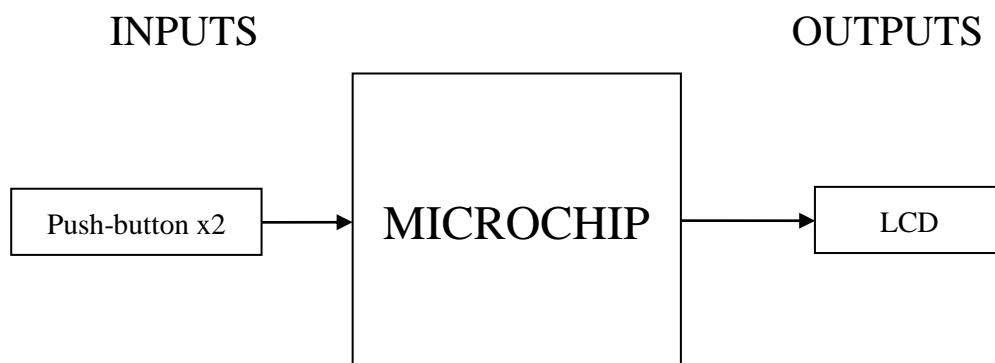


Figure 1: Inputs and Outputs

Design

Begin a new STEMSEL project in ezCircuit and add a LCD and a push-button to the circuit board. Note that the components should be inserted into the slots which appear as light green. All the components can be found in the circuit group.

The completed design should appear as shown below, where “Hour” is the push-button.

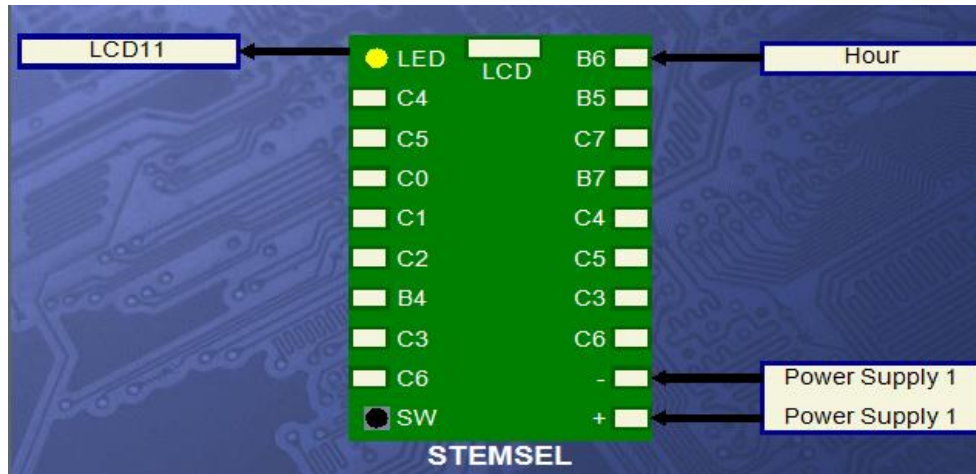


Figure 2: Circuit Design

Build

Insert the white signal wire, red and black wires into B6, positive (+) and negative (-) slots respectively. Then plug the LCD into the LCD port on the top of the circuit board. The complete circuit should appear as below.

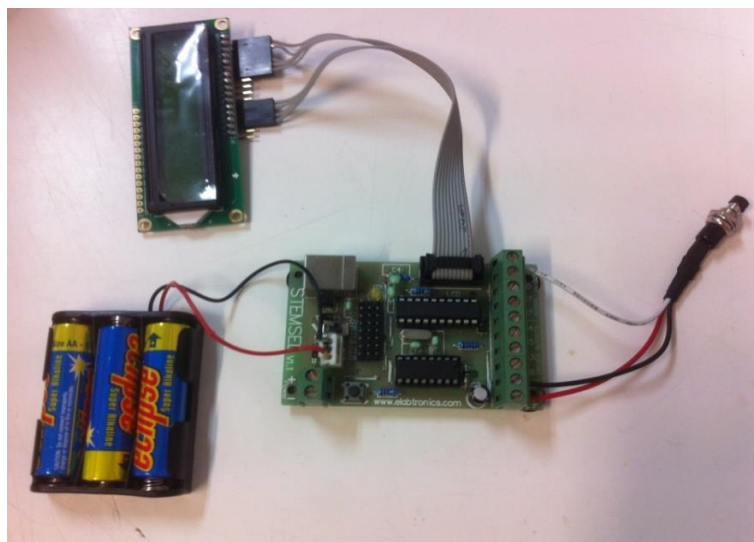


Figure 3: Circuit

Programming

With the design and assembly completed, send the design to CoreChart to program the microchip. Make sure the button is pushed when connecting the STEMSEL controller board to the computer. Before starting to program, make sure the circuit board and the components are tested using the testing program.

1. Our first step is to use two SetNumber icons create our variables that will store our minutes and hours. We will be starting at 12:00 midnight, so our hours are set to 12, minutes to 0.
2. We will not want to reset these every time we loop back, so add an address called Loop.
3. Now we will display the time. Add an LCDFormat icon, and use it to clear the LCD with no delay.
4. Use two DispNumberLCD icons to show Hours and Minutes on the LCD, with a LCDMessage icon in between to display a full stop to separate hours and minutes.
5. So that we don't have to wait for a full 12 hours to check our clock is working correctly (otherwise we would need to watch the clock tick for 12 hours...), we will use a shorter time delay of 10 hundredths of a second.
6. Now, increment the minutes variable using an Add icon to add 1 to Minutes, saving the result back as Minutes. Our clock will now count up by one minute every tenth of a second, or one hour every 6 seconds.
7. Use a Compare icon to see if Minutes is above 59. If it is, use an Add icon to add one to Hours, and a SetNumber icon to reset Minutes to 0. Group these together into a group called Increment_H so they both fit below the decision.
8. Similarly, use another Compare icon to see if the Hours variable is above 12. If so, then reset Hours to 1.
9. Finally, add a GoTo icon to return to the Loop address added earlier. Send this program to chip and test it. Note that due to the short time delay used, the display may flicker.

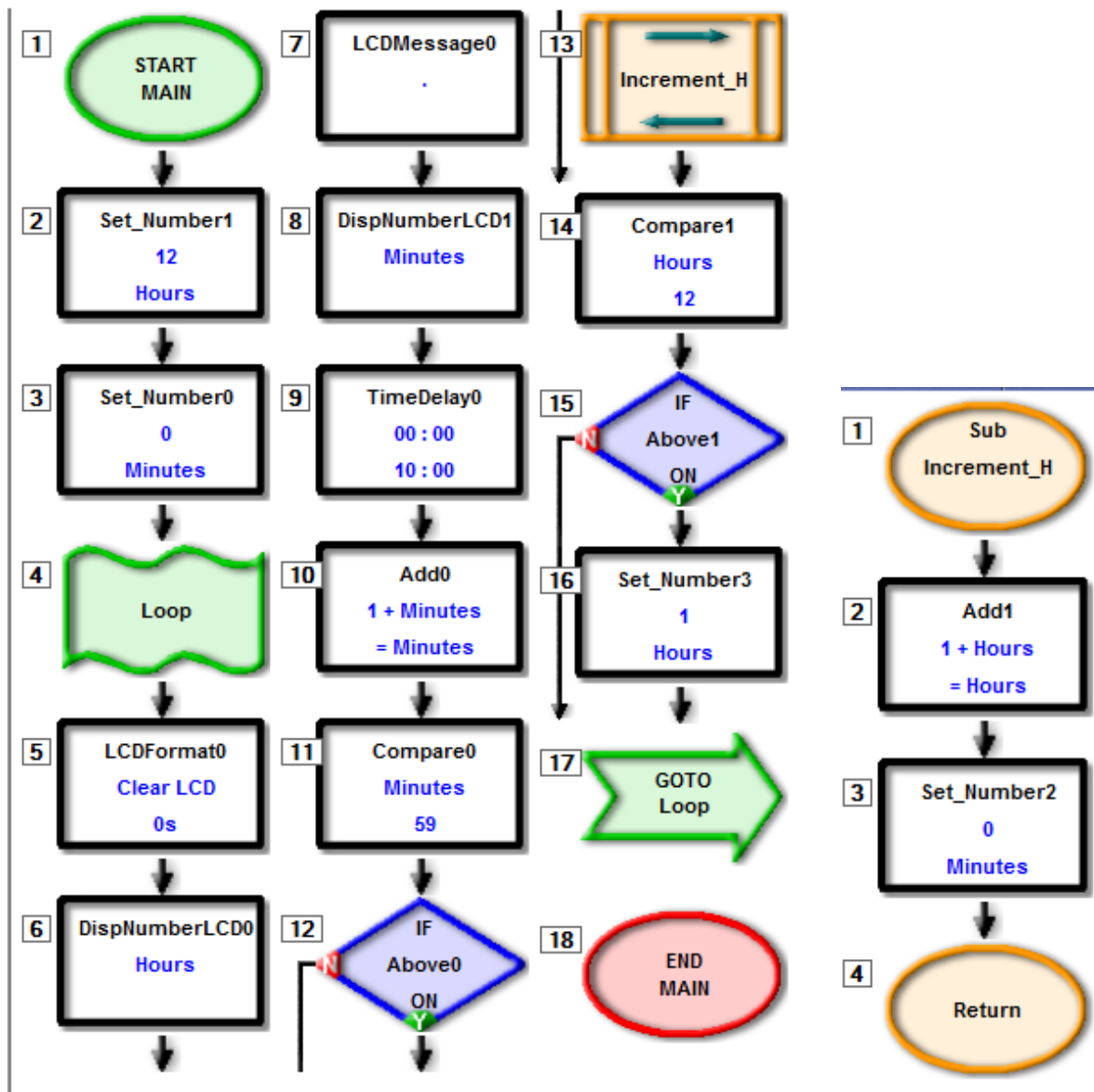


Figure 4: basic clock program

Next, the program should be able to set the clock using two pushbuttons.

1. First, group all the display icons (clear LCD, DispNumberLCD, LCDMessage) into one group called DisplayTime. The group “DisplayTime” will be used later so it is possible to see the time while the clock is being set.
2. Group the Time Delay, Add Minutes and Compare Minutes into another group called “Increment_Min” add the group “DisplayTime” inside the newly created Time Delay and return back to the main core chart.
3. Inside the group of Increment_H the program needs to check whether if the hour is greater than 12. Add a compare icon and compare “Hours” to 12 and if it is above 12, set the “Hours” back to 1.

4. To check if the Hour button is being pressed by using another group icon. First add a decision icon to check whether the button is being pushed or not. Add a time increment of 50 hundredth of second and add another “DisplayTime” group icon. Merge the time increment and DisplayTime together and call it “Set_Hour”.
5. Now we need to be able to set the minutes, using the decision icon we want to check whether the minute button (A3_Pushbutton currently onboard) is being pushed if the push button is OFF, we want to Increment_min and DisplayTime and add a 50 hundredth of a second time delay.
6. Group the Increment_min, DisplayTime and time delay into a group called “Set_Minutes”. Because we don’t want to double the minutes we will need another Decision Icon. If the A3_Pushbutton is OFF we want to go back to the subroutine that was just created “Set_Minutes”.
7. Send this new program to the chip and try setting the time.

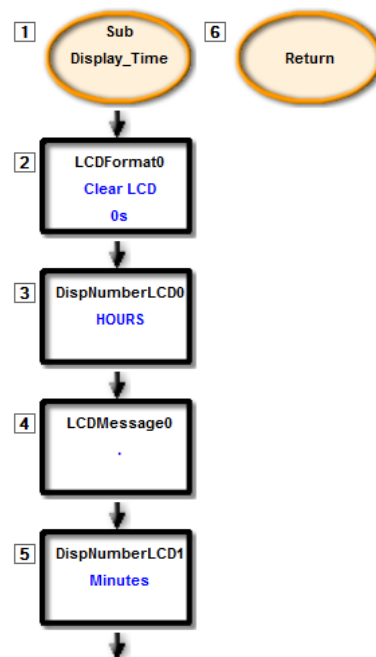


Figure 5: Display Time Subroutine

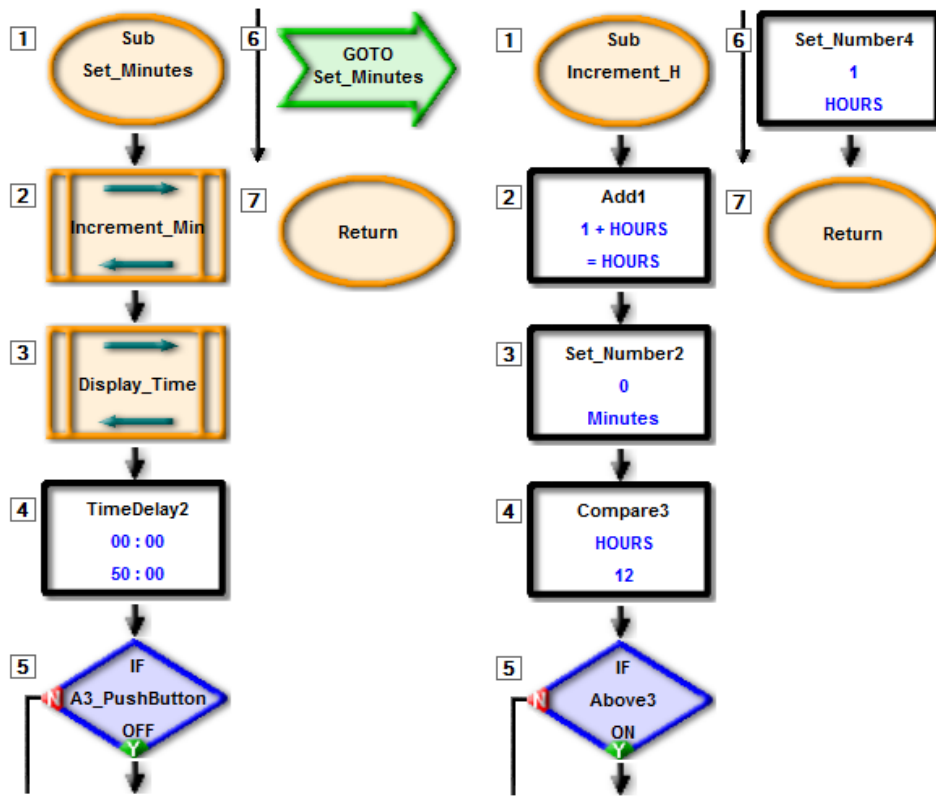


Figure 6: Increment_H and Set_Minutes Subroutines

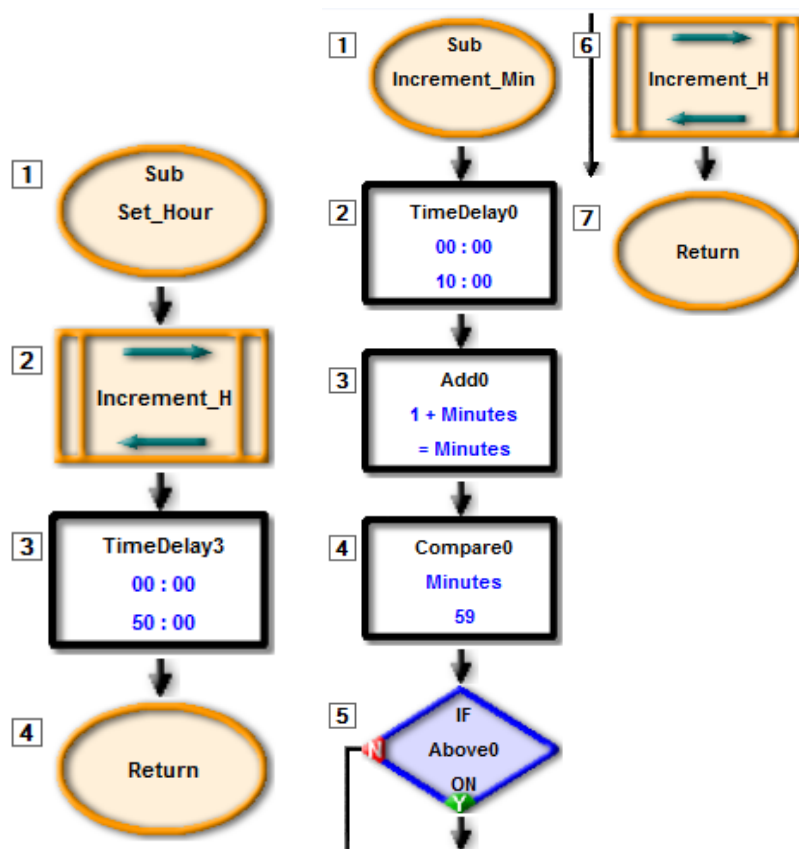


Figure 7: Set_Hour and Increment_Min Subroutine

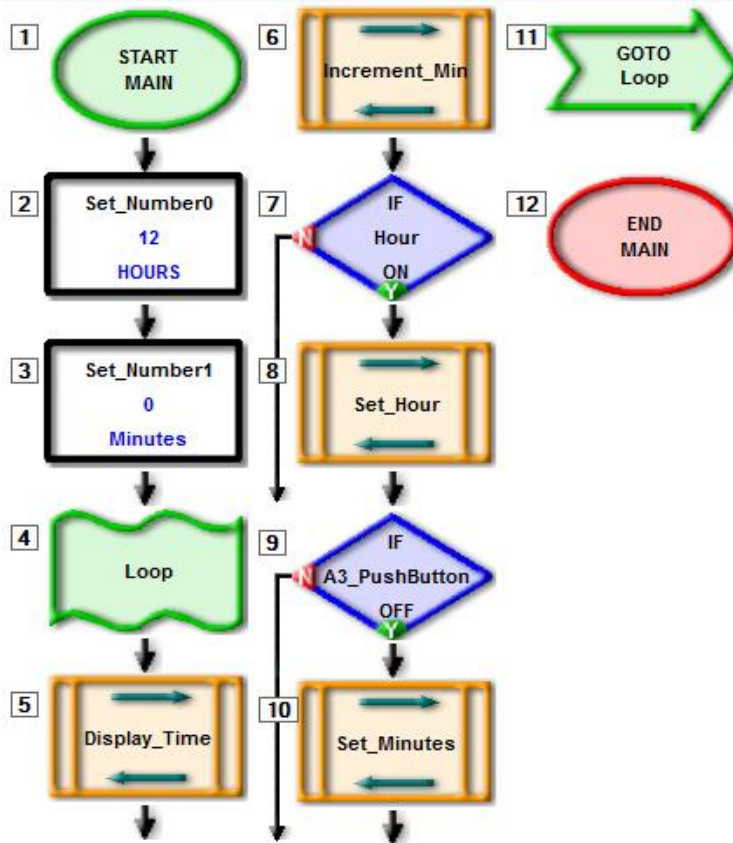


Figure 8: End Program with setting functions

Extension

There are many changes we could make to our clock, such as making it display in 24-hour time, displaying seconds, or adding a buzzer to make it an alarm clock. See how you go implementing one or even all of these ideas.

NOTE: seconds is a variable already used for the TimeDelay icon, so cannot be used for your seconds variable. Name your seconds variable “Sec” or something similar.

Summary

In this project we programmed a microchip which used a push-button as an input, and would activate a LCD to perform the operation of a clock. After this project, we should have a better understanding of how a clock works. Also the project should help develop our skills on programming the STEMSEL controller board using ezSystem.

At the end of the exercise, you should be able to:

- Manipulate the ‘add’ and ‘subtract’ functions in CoreChart.
- Use a LCD to display information in different formats.
- Use a push-button to control the program.