

The potentiometer will be the control switch which can be used to change the message displayed. Since the potentiometer can create different voltages, they will be used to decide the message displayed. Start with 3 different messages. The output range from our microchip potentiometer is 0 to 255. Therefore, the first message can be displayed when the output is from 0 to 85, display the second message for 86 to 170 and the third message for 171 to 255.

Design

According to the plan, the potentiometer will be needed as an input, a LCD board as display output and a STEMSEL controller board as a decision module.

Now it's time to open ezCircuit Designer to add these components, the final design should look like this:

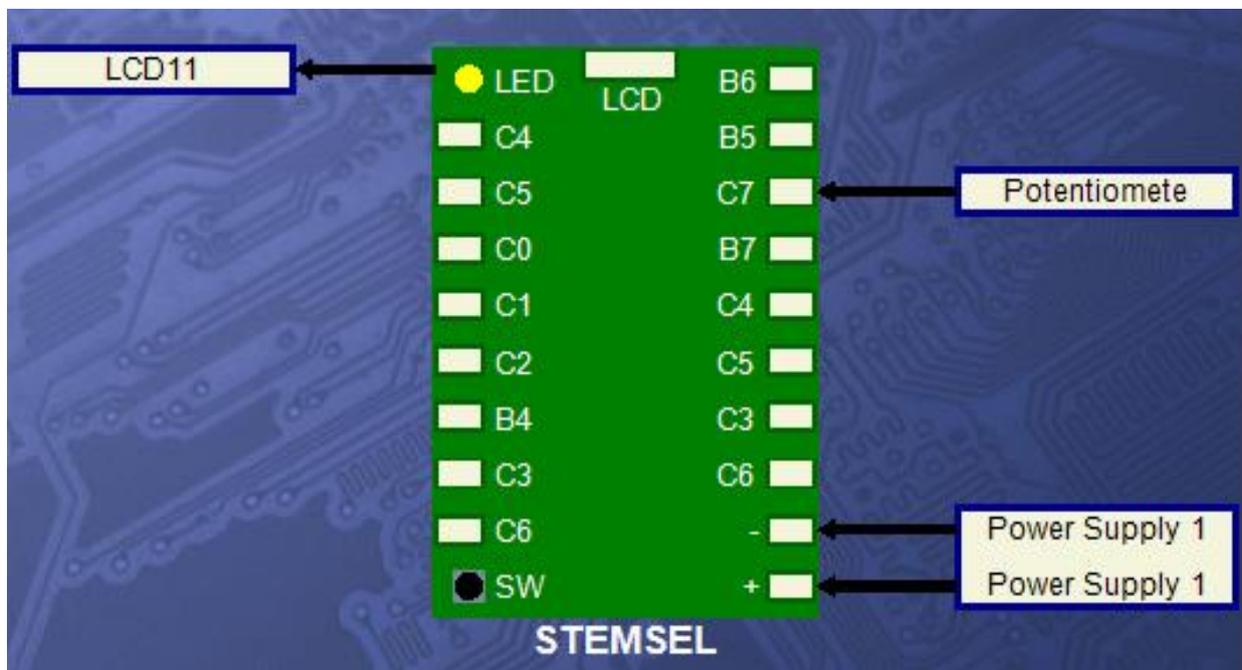


Figure 2: Circuit Design

Build

When the design has finished, it's time to assemble the circuit using the kit. Put all of the black wires into the negative (-) port and insert the red lead of the potentiometer into the positive (+) port. The white wires go into the ports specified in the design.

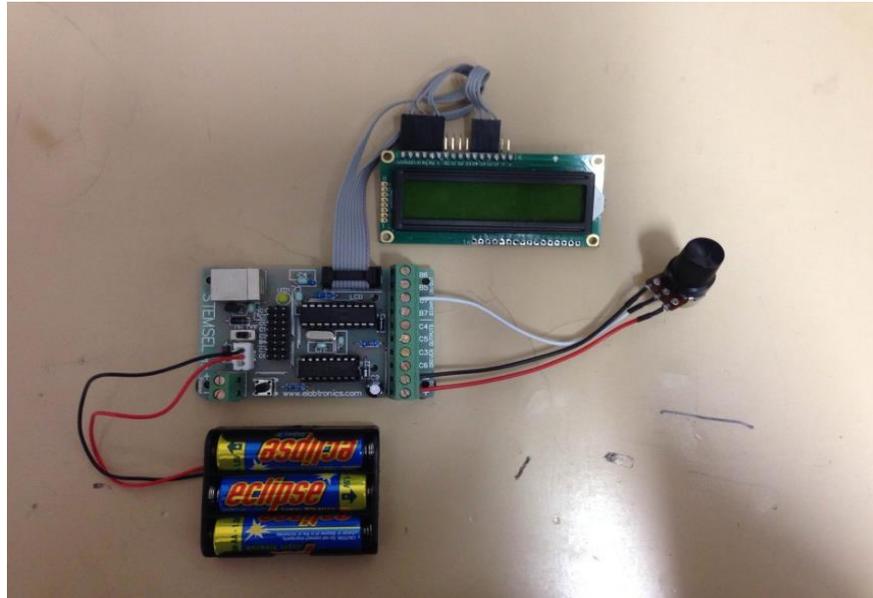


Figure 3: Circuit

Programming

Once the circuit has been assembled, send the design to CoreChart by clicking the “Send to CoreChart” button, then click the “Send Program To Chip” button to program the chip with the LCD test program. Once verified the LCD is working, delete the test routines *so we can start writing the program*.

1. As in the plan, a range of voltages need to be set for the messages. To begin with, Add an "Analog_In" underneath the "START MAIN" and save as "Number" .
2. A CompBetween icon can be used to compare the number with a range of voltages. Click the “Between” checkbox, so the program will check if the input is within the range or not. Remember that the POT outputs numbers from 0-255. Thus for three messages, there would need to be 3 ranges that the messages can fall under.
3. If the input is within the range, display the message on LCD by using the LCDMessage icon.
4. Start with displaying the three messages, repeat step 2 and 3 for other two messages.

5. After it has displayed, we want the LCD to hold the message, not just flash and disappear. Therefore, put a time delay at the end to hold the message.
6. In order to display the next message, the current LCD display first needs to be cleared, this can be done by using “LCDFormat”.
7. The program should be run as a loop so it can change when it needs, to do this a “GOTO START” needs to be used at the end.

The program should look like this:

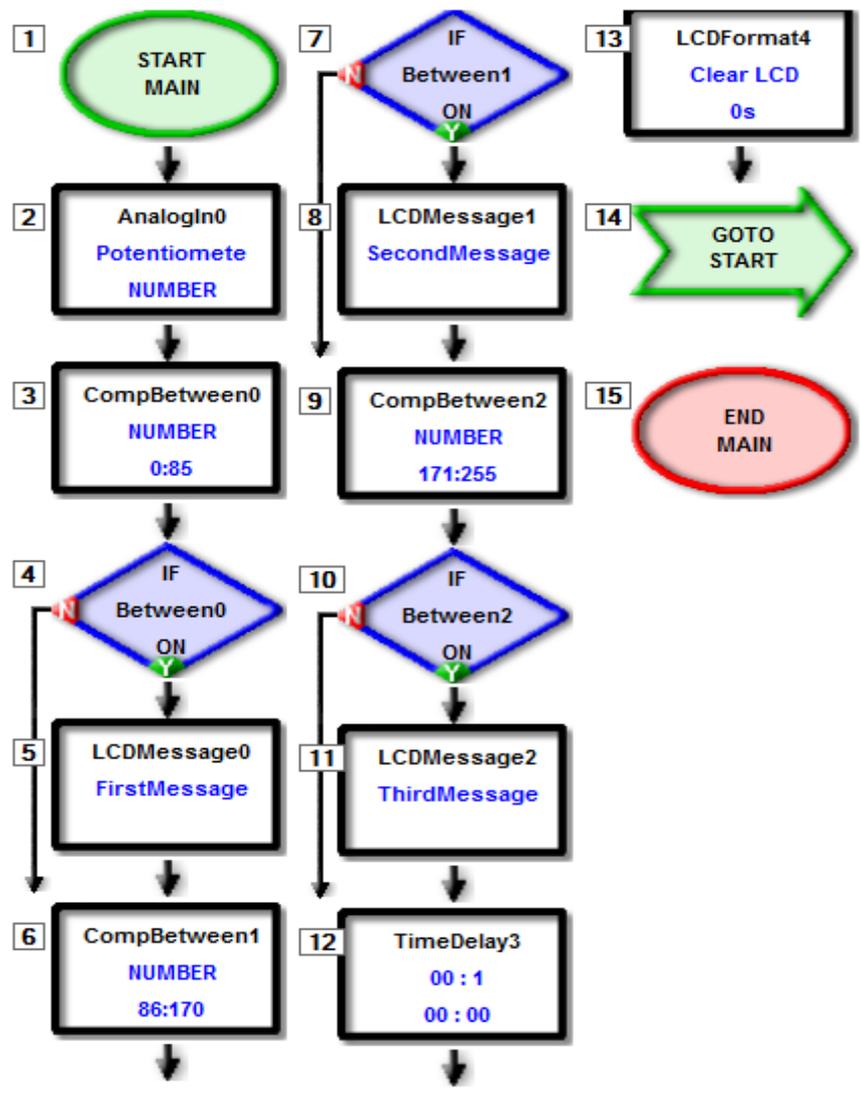


Figure 4: Code

After the program is sent to the chip, the potentiometer can be adjusted to display messages. In the example program, when the potentiometer input is between 0 and 85, “FirstMessage” will be displayed, when the input is between 86 and 170, “SecondMessage” will be displayed, and from 171 to 255, the message will be “ThirdMessage”.

Extension

If the display board works properly, change the message and invite your friends to give ideas. This display board can be used as a traffic light to display: “red”, “green” and “yellow”. Alternatively it can be used as a progress status report by displaying “Starting”, “Detecting”, and “Finishing”.

Also the code can be longer by increasing the number of messages that can be displayed. Then the user can figure out how many messages this display board can display, because when the number of messages increases, the range of voltages for each message will decrease. At a certain point, it will become difficult to select the message.

Instead of using 0 to 255, it can also use the voltage range of 0 to 5V. The most anticlockwise position corresponds to 0V while the most clockwise position corresponds to 5V. Change the code, see if it works and which method is better.

Summary

By using a microchip, a simple potentiometer and LCD can make a display board display messages. During this project, we learned the basic structure of a message display board, and how to use the CompBetween icon to compare a value with a range of numbers. It is simple, and can be developed into many different fields.