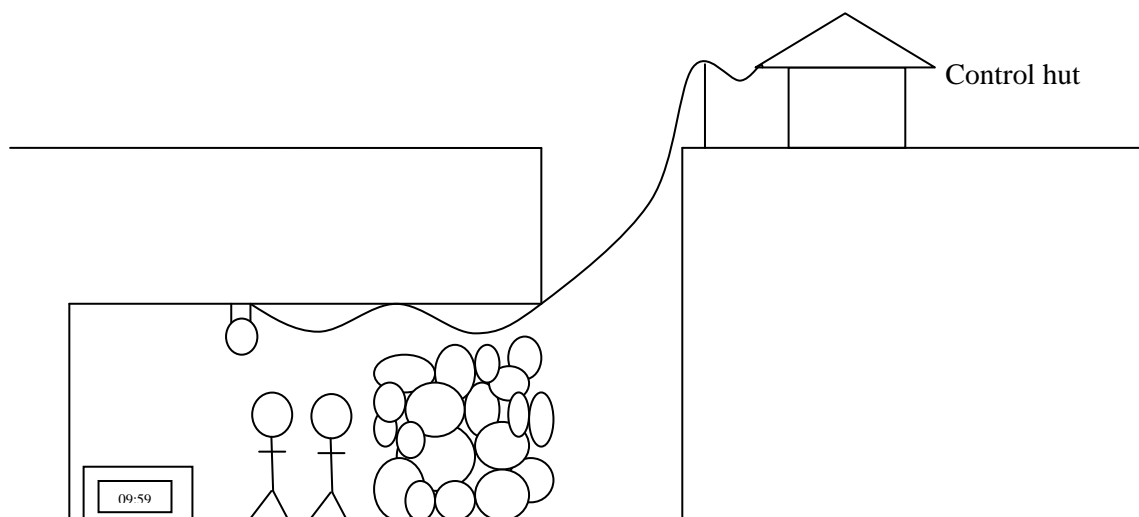


STEMSEL Intermediate Project 1: Trapped Miners

Problem

Consider a scenario where miners are placing an explosive charge to fracture the rock and make the mine bigger. Suddenly there is a cave-in, trapping the miners with the explosive! There is a programmable microchip on the surface which is connected to the light in the cave visible to the miners. The only way the miners will remain alive is by diffusing the explosive but the code is unknown to them and this is where you come to their rescue! The only way to communicate with the miners is using this light. Use your superpowers of programming to convey the 4 digit defuse code!



Background

What is a microchip? Micro, as you may know, means very small, and a chip is a piece like a potato chip or a wood chip. In the case of microchips, it is a small piece of memory that we can use to store instructions to tell other electronic components what to do, for instance. We can program microchips to do all sorts of different things, and they are especially good at tasks that are repetitive or require precise timing.

Ideas

How can we send the code to the miners using just the light? How can we ensure the miners know when one number is finished and we have started to send the next number? Should we send the code just once or multiple times?

Plan

We will use a sort of simplified Morse code to send each number to the miners by flashing the light on and off. In order to send numbers, we only need to blink the light the same number of times as the number we want to send, so to send a 4 we would blink the light four times. After one number has been sent, there should be a short delay before we start sending the next number. At the end, there should be a long delay before the entire code is repeated in case the miners didn't notice the first time.

Please Note: An example code [3 4 1 8] has been used for demonstration purposes and students are required to come up with a 4 digit code themselves.

Design

The design of the micro-controller should incorporate the lightbulb in the cave. Firstly, open the ezCircuit Designer and begin a new STEMSEL project and then add a light (use the Driver icon) by choosing it from the circuit group.

The completed design should appear as shown below:

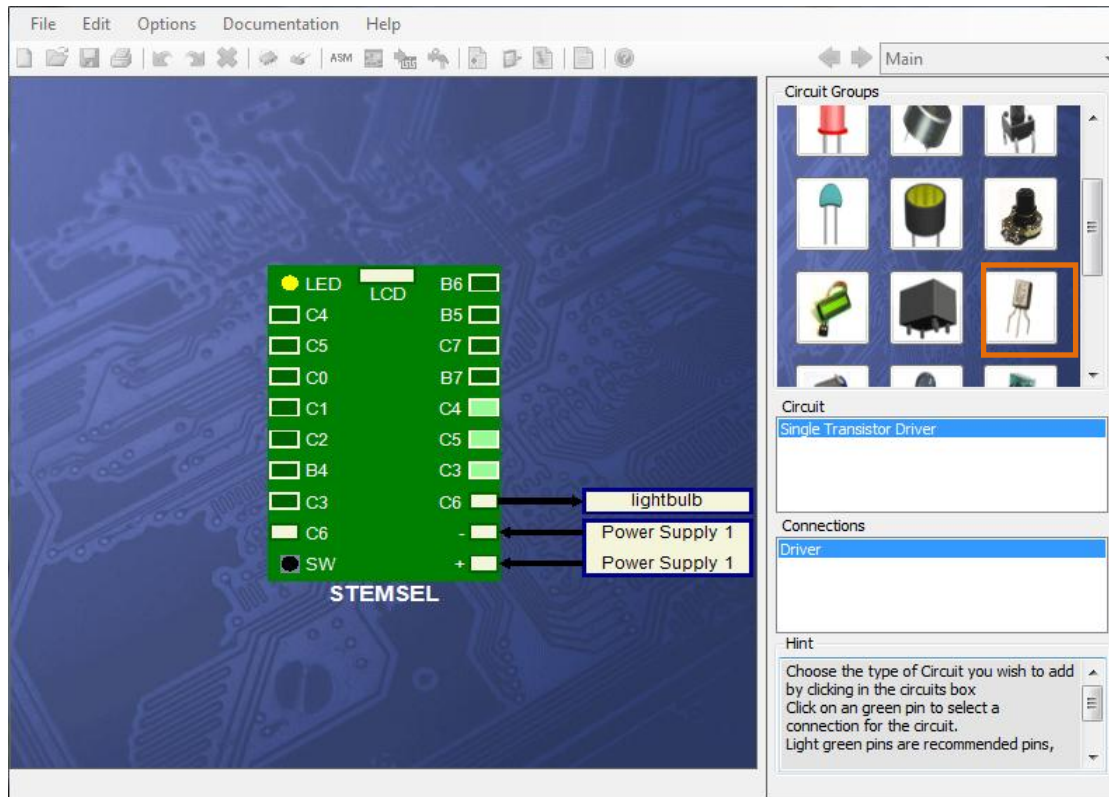


Figure 1: Circuit design

Build the circuit

Now that the design is complete, attach the black wire of the lightbulb to the slot on the STEMSEL controller board which has a negative (-) sign and the white wire to the slot C6.

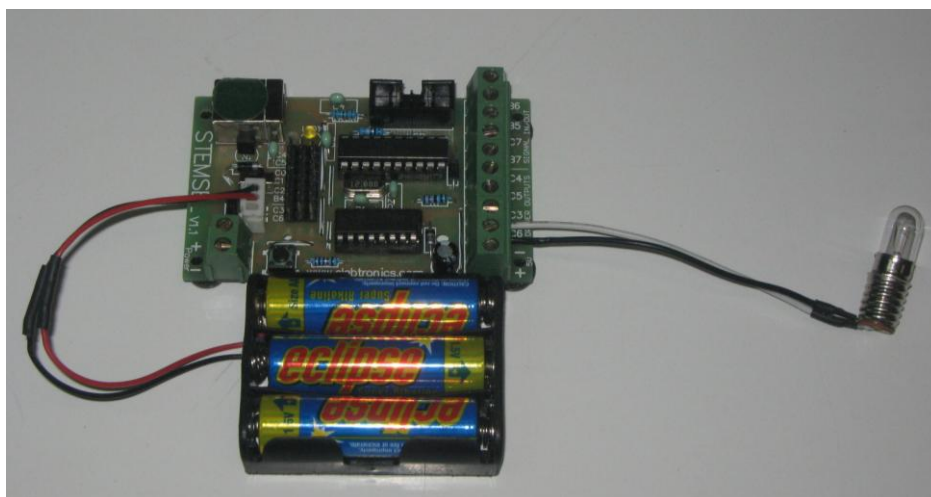


Figure 2: circuit

Programming

With the design and assembly completed, the design should be exported to Corechart to program the microchip.

In this relatively simple application, each number in the code refers to the number of flashes or blinks of the lightbulb in the cave, just like we said in the plan. One blink can be written as a sub routine in this code. Follow the steps below to write this sub routine:

1. In the 'ezSystem Mode Icons' panel in the far right end of the window, click on the 'Outputs' icon. Under this, select 'OnOffPin' and then click on the arrow right after the 'START' icon. Double clicking on this newly appeared 'OnOffPin' icon allows you to select 'lightbulb' from the OUTPUT PIN drop down menu. Ensure 'On' is selected in this window.
2. Now, to make the lightbulb stay on for a brief time, a delay needs to be added. Click on the Numbers icon on the ezSystem mode icons and select TimeDelay in the Icon Properties field. Now insert the delay icon in the program then customise it to 0.5 seconds, i.e. enter 50 in the hundredths place.
3. After the 0.5 seconds, the lightbulb must be turned off. So insert another OnOffPin icon in the program and customise the lightbulb to off.
4. Now add another half second time delay after the lightbulb is switched off, this will lapse some time before lightbulb turns on again.
5. Now that the subroutine is done, group all the icons together and give it a meaningful name. For example, in the current project it is named as "Blink"

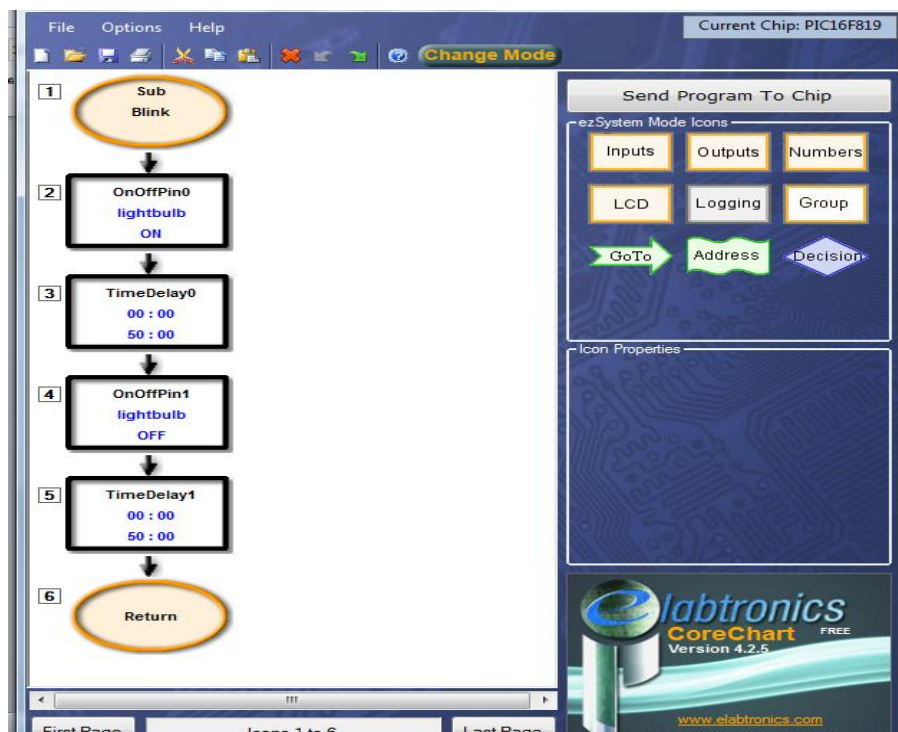


Figure 3: The Blink group

The first number in the example code is 3 therefore the light should blink 3 times. For that to occur, you need to have two more blinks which can be added by a simple copy and paste of the subroutine written above.

In the plan, we said that there should be a delay between each number so that the miners could see the end of one number and the start of the next. Between each number in the code, a time delay of 3 seconds should occur to mark the end of one number and the beginning of the next number. Applying the procedure set out in step 2 add a time delay of 3 seconds at the end of the first set of blinks. Continue adding Blink subroutines and time delays to make your code. According to the plan, at the end of the code we should use a long delay to highlight that the whole code has been sent. Again, follow step 2 to add a 10 second time delay just before the end of the code, then use a GoTo icon to send the program back to the start in order to send the code again.

Check to ensure your completed code appears like shown below.

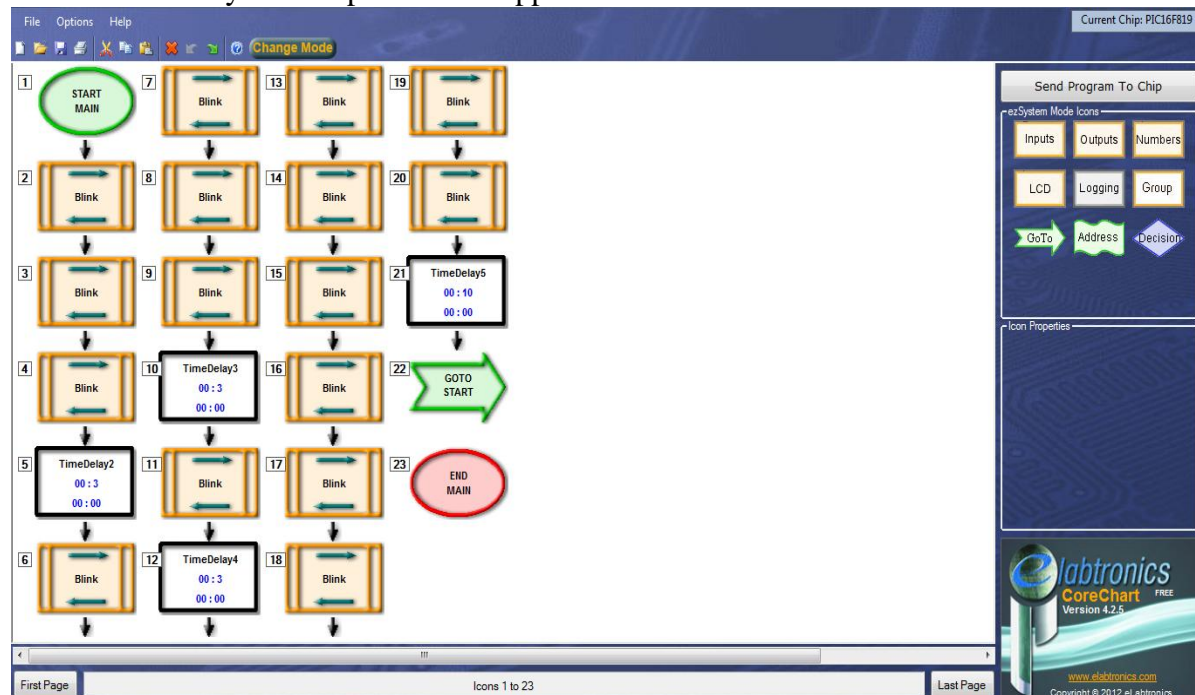


Figure 4: the completed program

Once you have finished, create your own code and see if other people can understand your code.

Bonus question

What if there was a zero in the code? How would you send a 0 on your transmitter?

You can't just leave the light off to send a 0, since it is likely that it will be mistaken for the delays between numbers or at the end of the code. Instead, it would be better to send ten flashes since this will be different to both the other numbers and the delays between them. Can you think of another way to send a zero?

Summary

If given a boring repetitive task, a human would become bored and possibly make mistakes. A microchip on the other hand never gets bored and can continue doing the task all day. Because of this, microchips are very good at sending signals repeatedly. The simple technique of flashing the lightbulb and adding the time delay can be incorporated into many tasks of high importance like in this case, delivering the code to diffuse an explosive. At the end of this exercise, you should be able to:

1. Manipulate the 'time delay' function in the core chart
2. Use the technique of flashing a lightbulb to display information.