Morse Code Generator

Guided Project Instructions

Figure 1 Morse
Telegraph Key

Related Core Concepts:

Solenoid

Microcontroller Simulator

Learn It!

Morse code is a simple means of transmitting characters and messages using a single tone whose duration is modulated. The code is a standard used internationally and encompasses more than just the English alphabet. A single character is comprised of short tone impulses called "dots" and longer held tones called "dashes". The timing of the dots to dashes is 3 to 1 with a pause between dots and dashes equal to the length of a dot. There is a pause time between letters that is the same length as a dash. Morse code can be transmitted over different mediums like sound, light and radio frequencies. In this module you will create a type of Morse code generator, replacing a light or sound with a solenoid. This modification introduces new challenges like how to solve the timing issue between dots and dashes and how to properly time pauses between dots and dashes. This module allows you to create an elegant solution to transmitting information and the techniques you implement can be carried over to a large communications system.

"This module allows you to create an elegant solution to transmitting information and the techniques you implement can be carried over to a large communications system."

Build It!

Morse code is based off of properly timing Boolean outputs. The main challenge is to take a character input from a user and converting it to its Morse code equivalent pattern. Once the pattern for the character is chosen the proper timing for true and false levels needs to be decided on. This information can then be used to directly control a solenoid circuit.

Task 1: Build a solenoid circuit that will allow for it to be controlled with a low current digital signal. Tie the input of this circuit to one of the digital lines of your National Instruments hardware.

Task 2: Create LabVIEW code that reads a user letter input and outputs an array of Boolean values that match the Morse code representation of the character. You can use an array lookup table to produce your output. Use a single Boolean true for a dot and two consecutive Boolean true values to represent a dash.

Task 3: Modify your LabVIEW code to instead use an array lookup table of time delays for dots and dashes. Since we are using a solenoid we cannot simply have a longer delay time for the dash since this may confuse a user. Create a solution that will still give a clear delimitation between the dots and dashes but allows for easily maintaining time. The result from this should be for the user to input a letter and get returned a 1D array with times for the Morse code pattern for that letter.

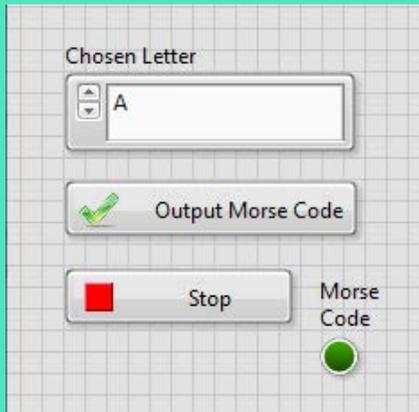Figure 2 Solenoid set up to tap against a heavy metal surface

Figure 3 Example Morse code front panel

Task 4: Modify your code to parse the array of delay times and use them to blink an LED. For example, the LED should turn on and stay on as long as stated by the first value of the array then turn off and stay off for as long as stated from the second value in the array and so on.

Task 5: Drive the digital output of your National Instruments hardware with the Boolean pulse train you created from the previous step. Tie this Boolean output to an LED to verify the LED blinks as you would expect. Connect this Boolean output to the solenoid circuit once you are finished testing.

Guiding Questions:
- Is there a limit to how fast you can toggle the mechanical action of a solenoid? What are the concerns with using a solenoid.
- Can the Morse code solution be realized with polling? What is another means of pulling data into your code?
- Explain hardware and software timing using the National Instruments hardware. What constraints on timing does your hardware present?

Expand it!
- Expand your code to allow for words to be input by users. This should pay attention to the timing between characters but should be a replication of the current code. This could possibly be realized using a For loop.

Research It!

Morse Code and the Telegraph

http://www.history.com/topics/inventions/telegraph

LabVIEW Timing

http://www.ni.com/tutorial/7592/en/