Projects: Build a Sensor Control Program

Learning how to use LabVIEW is best done by actually using LabVIEW. Reading or hearing lectures can be helpful at first, but actually starting a LabVIEW project from scratch, designing a front panel and block diagram, and figuring out the steps and logic required to perform a task is how you will really learn LabVIEW.

Below are two projects. Read through the description and design requirements of each project, and then choose one project to work on. If you have time, do both.

PROJECT 1 – AUDIO FEEDBACK

The LabVIEW front panel is used to design a user interface that provides visual feedback in the form of charts and graphs, meters, thermometers, flashing LEDs, and other indicators. Audio feedback is an alternative method that may, in some circumstances, be preferred. Examples of audio feedback include smoke alarms that emit an audible tone when smoke is detected, stoves that beep when a pre-defined temperature is reached, and manufacturing stations that provide an emergency sound when corrective action is required. Often, it is not important to know the exact value of a measurement, but only an approximation or a change in the value.

DESIGN REQUIREMENTS

In this project, you will create a program that provides an audio signal as feedback of temperature. You will use a Temperature Probe to measure the temperature in degrees Celsius. The sound should be created by your computer, with the frequency of the tone directly proportional to the temperature reading.

MATERIALS

SensorDAQ or LabQuest interface	LabVIEW
Vernier Stainless Steel Temperature Probe	USB cable
computer	beaker of warm tap water

TIPS

1. The LabVIEW functions palette includes a Graphics and Sound subpalette. Inside this subpalette is a subVI called Beep.vi. Use the Beep subVI as the means to making sound with your computer.

2. A good way to start is to build a simple program to get to know how this subVI works. Place it within a While Loop and control the frequency. Create constants for the “duration” and “use system alert?” input terminals. The “use system alert?” terminal must be set to False so that the sound is based on your frequency value, rather than the system alert sound. Use this VI to determine what upper and lower level frequency values are appropriate in terms of audible feedback to the user.

PROJECT 2 – CELL PHONE DECODER

Cell phones continue to add features that increase their importance to people throughout the world. Communicating with family, friends, and business associates are common uses, and providing a means for emergency contact may be the most important feature of all. Cell phones are also capable of wireless internet access, sending and receiving photos and files, GPS receiver, MP3 player, and even sensor feedback. But the original feature of a cell phone, as well as a land-line phone, was the ability to transmit numbers, symbols, and letters to the telephone company with the push of a button.

Dual Tone Multi-Frequency, or DTMF, is a communication method developed by engineers with the trademark name “touch-tone,” and allows the telephone company to know what numbers you dial, issue commands to a switching system, or enable the caller to communicate with a computer system using the numbered keypad.

The DTMF system uses eight different frequency signals transmitted in pairs. Each time you press a key, two tones are produced. For example, when you press “5,” your phone produces a 770 Hz sinusoidal signal and a 1336 Hz sinusoidal signal. The following table shows how the DTMF keypad is laid out in a 4×4 matrix, with each row representing a low frequency, and each column representing a high frequency.

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

In this project, your challenge is to create a program that is able to decode a DTMF signal. If you accomplish this goal, your program should be able to tell you what button was pressed on a cell phone out of your sight by decoding the waveform, as the telephone company does.

DESIGN REQUIREMENTS

In this project, you will create a Dual Tone Multi-Frequency (DTMF) decoder for a cell phone or tone dialer. If these are not available, a third option is to run the Cell Phone.vi program (found on the accompanying CD), and use the speakers of your computer to generate the tones. You will use a Microphone to measure the waveform of sound as a user presses the buttons from 1 to 9. Your program should display the value of the selected number for as long as the phone, tone dialer, or computer is producing a noise. The number should no longer be displayed when the noise from the button press ends.

MATERIALS

SensorDAQ or LabQuest interface
USB cable
computer
LabVIEW
Vernier Microphone
cell phone, tone dialer, or computer running Cell Phone.vi

TIPS

1. Consider creating your program in steps. The first step is to continuously read the microphone data at a fast sampling rate. The next step is to provide waveform analysis to determine the low frequency and the high frequency within the signal. The final step is to provide decision making within your program to display a number that corresponds to the low and high frequency.

2. Consider the Tone Measurements Express VI as a way to search a specified frequency range to find the single tone with the highest amplitude.

3. Express VIs typically provide results that are passed as Dynamic Data. You may have to convert this to a numeric data type. To do this, you will need to use the Convert From Dynamic Data Express VI. Choose the Single Scalar option from the configuration dialog box.

4. Your frequency measurements are not likely to match exactly the numbers shown in the table above. Case structures allow you to specify a range of values for comparisons (for example, rather than specifying a value of 697, use 685..715 instead).