

Software Lab (EEP702)

Assignment 4

Name : Dushyant Sahoo
Entry Number : 2011EE50546

Department of Electrical Engineering



Indian Institute of Technology Delhi

Feb 24, 2015

Contents

1 Abstract:

This assignment contains a Graphical User Interface that resembles a full-fledged calculator. This calculator supports all the basic functionalities like- BODMAS, natural log, exponential, square, square-root, and many more. The calculator supports the integers as well as decimal point input and output. The text-to-speech conversion is done for both, the integers and double-type decimals. The calculator designed is much interactive in terms of operation as compared to the conventional calculators. The property of speech differentiates it from the trivial designs. The display of the calculator is much similar to the original hardware-calculators in a way that it displays both, the input and the corresponding output as well, for a particular expression calculation.

2 Problem Statement:

Design a user interface(GUI) for scientific calculator in Qt. The calculator should evaluate all the basic functions,viz. : (+,/,*%,sin,cos,tan, \log_{10} , \log_e ,square root,square). The user will be asked to enter an expression using buttons like we have in our calculator. Entered expression should be shown in a box as an when the user is entering the expression (say, 65+23 or 56.87+56 or ()+45 or cos(45)) like we do normally in our scientific calculator and will be provided a button named calculate. On pressing this button(Follow standard associativity and preferences) , the final result of that expression has to be displayed in numbers and words in two different boxes. There should also be a button, after clicking which the final result calculated has to be spoken. This can be done using libraries in Qt.

3 Introduction:

3.1 QT:

- QT is a cross-platform application framework that is widely used for developing application software that can be run on various software and hardware platforms with little or no change in the underlying codebase, while having the power and speed of native applications.
- Qt is used mainly for developing application software with graphical user interfaces (GUIs); however, programs without a GUI can be developed, such as command-line tools and consoles for servers. An example of a non-GUI program using Qt is the Cutelyst web framework. GUI programs created with Qt can have a native-looking interface, in which cases Qt is classified as a widget toolkit.
- Qt uses standard C++ with extensions including signals and slots that simplifies handling of events, and this helps in development of both GUI and server applications which receive their own set of event information and should process them accordingly. Qt supports many compilers, including the GCC C++ compiler and the Visual Studio suite. Qt also provides Qt Quick, that includes a declarative scripting language called QML that allows using JavaScript to provide the logic. With Qt Quick, rapid application development for mobile devices became possible, although logic can be written with native code as well to achieve the best possible performance. Qt can be used in several other programming languages via language bindings. It runs on the major desktop platforms and some of the mobile platforms.

3.2 BC:

'BC' stands for Basic Calculator, "an arbitrary precision calculator language" with syntax similar to the C programming language. BC is typically used as either a mathematical scripting language or as an interactive mathematical shell. 'BC' is often referred to as Bench Calculator.

- A typical interactive usage is typing the command `bc` on a Unix command prompt and entering a mathematical expression, such as `(1 + 3) * 2`, whereupon 8 will be output. While `bc` can work with arbitrary precision, it actually defaults to zero digits after the decimal point - so the expression `2/3` yields 0. This can surprise new `bc` users unaware of this fact. The `-l` option to `bc` sets the default scale (digits after the decimal point) to 20, and adds several additional mathematical functions to the language.

- GNU bc derives from the POSIX standard and includes many enhancements. It is entirely separate from dc-based implementations of the POSIX standard and is instead written in C. Nevertheless, it is fully backwards compatible as all POSIX bc programs will run unmodified as GNU bc programs. GNU bc variables, arrays and function names may contain more than one character, some more operators have been included from C, and notably, an if clause may be followed by an else. Output is achieved either by deliberately not assigning a result of a calculation to a variable (the POSIX way) or by using the added print statement. Furthermore, a read statement allows the interactive input of a number into a running calculation. In addition to C-style comments, a 'hash' character will cause everything after it until the next new-line to be ignored. The value of the last calculation is always stored within the additional built-in last variable.

3.3 BASH SCRIPT:

The Bash command syntax is a superset of the Bourne shell command syntax. The vast majority of Bourne shell scripts can be executed by Bash without modification, with the exception of Bourne shell scripts stumbling into fringe syntax behavior interpreted differently in Bash or attempting to run a system command matching a newer Bash builtin, etc. Bash command syntax includes ideas drawn from the Korn shell (ksh) and the C shell (csh) such as command line editing, command history, the directory stack, the \$ RANDOM and \$ PPID variables, and POSIX command substitution syntax \$ (). When used as an interactive command shell and pressing the tab key, Bash automatically uses command line completion to match partly typed program names, filenames and variable names. The Bash command-line completion system is very flexible and customizable, and is often packaged with functions that complete arguments and filenames for specific programs and tasks.

3.4 GOOGLE Text-to-Speech:

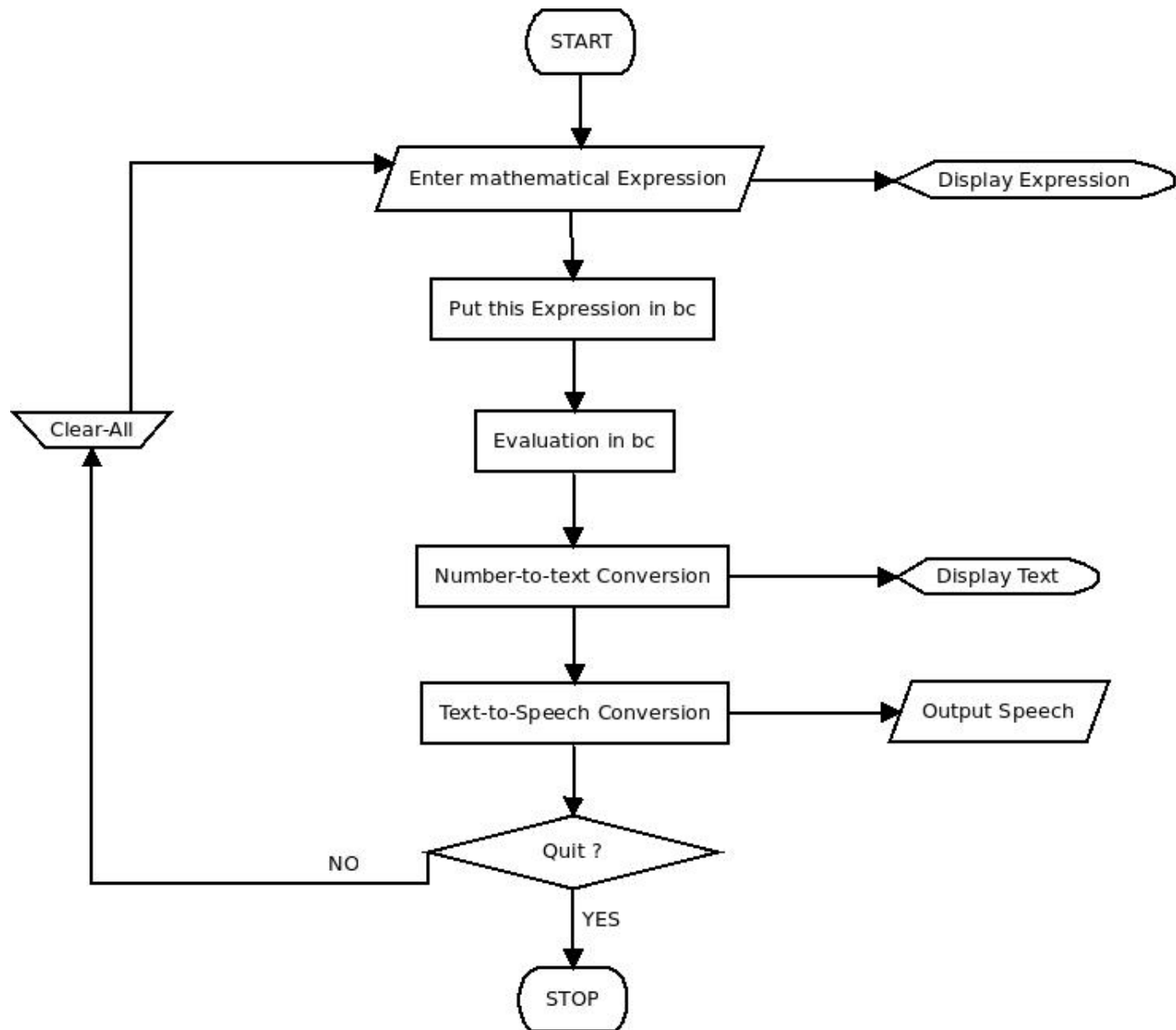
Googles Text to Speech engine is a little different from Festival and Espeak. The text is sent to Googles servers to generate the speech file which is then returned to our Pi and played using mplayer. This means that it will need an internet connection for it to work, but the speech quality is superb. The widget has used ax206geeks bash script to access the Google Text to Speech engine:

4 Methodology:

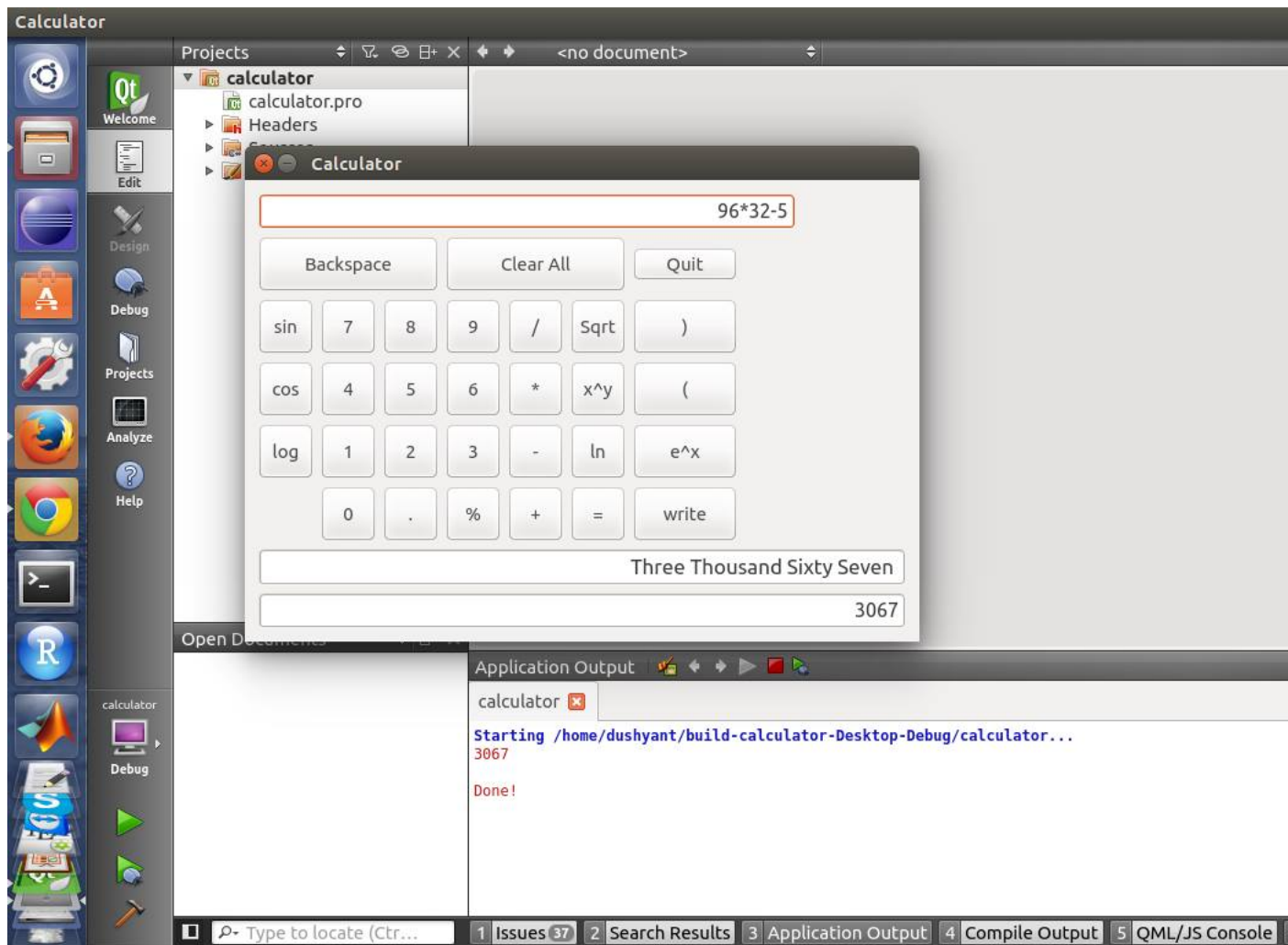
4.1 Brief analogy:

- Enter the valid mathematical expression in the input bar of calculator.
- Buttons are provided for user's convenience with various functionalities, and basic integers.
- After entering full expression, user should press 'equal to' button to evaluate the expression.
- The entered expression gets passed to the bash command-line calculator.
- The desired output gets displayed on the output bar of the calculator after full evaluation.
- The calculator also outputs the speech that conveys the output number in text/word form converted from the original number form.

5 Flowchart:



6 Screenshot:



7 Conclusions:

The implementation of the calculator is done using Qt creator, bash, and the bc tool of the command-line. The widget, full-fledged working model of a scientific calculator supporting various functionalities, is designed and implemented successfully.