Sensing and Instrumentation Practice

Scientific Calculator

Nehemiae G. Roy (EVD18I018)

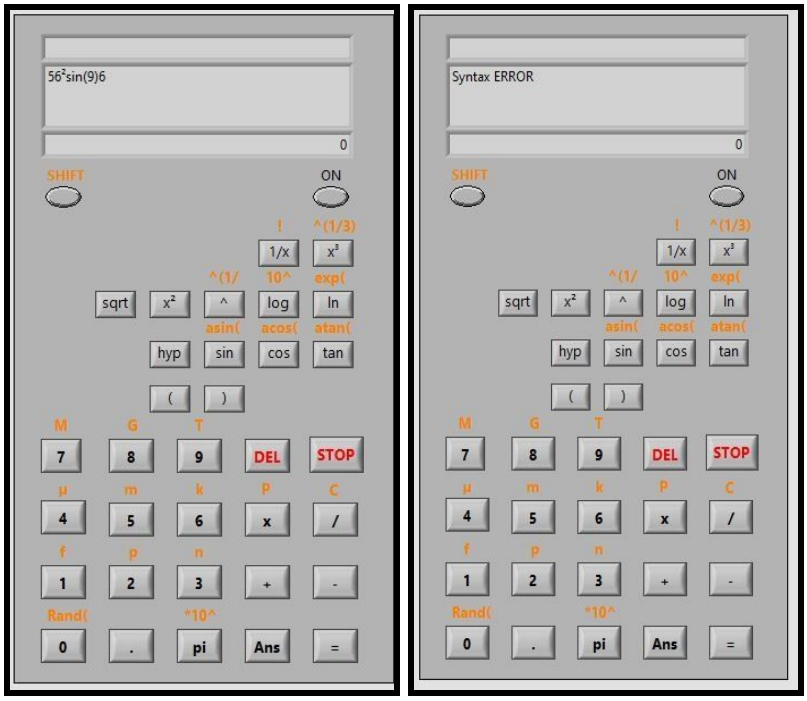
I. Explanation

The `Event Structure` can be used to trigger different cases after different events have occurred. This is precisely what a real time calculator requires and hence we can map common operations into a single case and then perform the manipulation. Specifically we have access to the `Property Node` of the button that triggered the VI. Using this we can make use of the `Label, Caption,` etc. of the button to write itself to the output screen. Shift registers can be used to initialize the displays and constantly update them. One of them holds the current input screen expression, another one the current output, and three others hold the state of the hyp, SHIFT and = buttons. These dynamically changing values can be used to determine what to display, what to calculate etc. We have made use of the `Eval Formula String` VI to calculate the corresponding value of the expression, but since many of the required operators are not present, I created another sub-VI `Search and Replace` to correctly adjust the input string in a suitable format for the former block - e.g. the engineering symbols need to be replaced with their actual values, permutation and combination calculations must be done and their result replace the input, etc.

The calculator is able to handle an entire expression (size limited by indicator size) and display the appropriate output.

One of the interesting things I have done is to use the DEL button to delete operators entirely just like in a calculator - e.g. sin(must be deleted in one shot while 567 must be deleted digit by digit.

By using the SHIFT button a host of other operations are available and can be used just like in a calculator. It is even capable of displaying errors in appropriate cases.



II. Features of a Scientific Calculator

1. Basic Calculations:
    a. Fraction Calculations
    b. Percent Calculations
    c. Degree, Minute, Second, Calculations
    d. Using Engineering Symbols
    e. Calculation History
    f. Scientific Constants
    g. Metric Conversions
2. Memory Functions:
    a. Answer Memory (Ans)
    b. Variable Memory (A, B, C, D, E, F, M, X, Y)
3. Function Calculations:
    a. $sin,\ cos,\ tan,\ sin^{-1},\ cos^{-1},\ tan^{-1},\ sinh,\ cosh,\ tanh,\ sinh^{-1},\ cosh^{-1},\ tanh^{-1}$
    b. $10^x,\ e^x,\ log,\ ln$
    c. $x^2,\ x^3,\ x^y,\ \sqrt{},\ \sqrt[3]{},\ \sqrt[x]{},\ x^{-1}$

d. $\int dx,\ \frac{d}{dx}$

e. Polar and Rectangular Coordinates conversions

f. $x!$

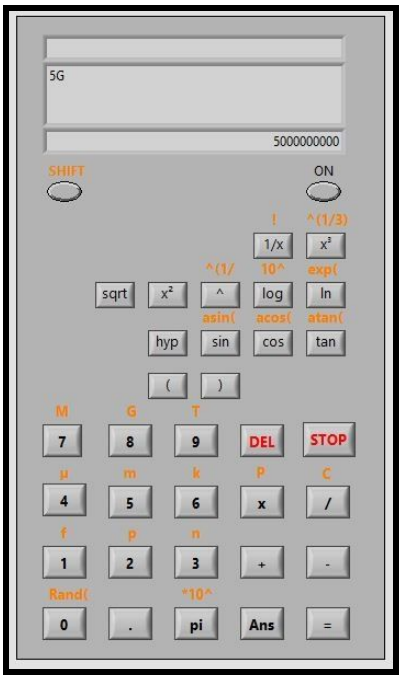
g. Random Number Generator

h. $nPr,\ nCr$

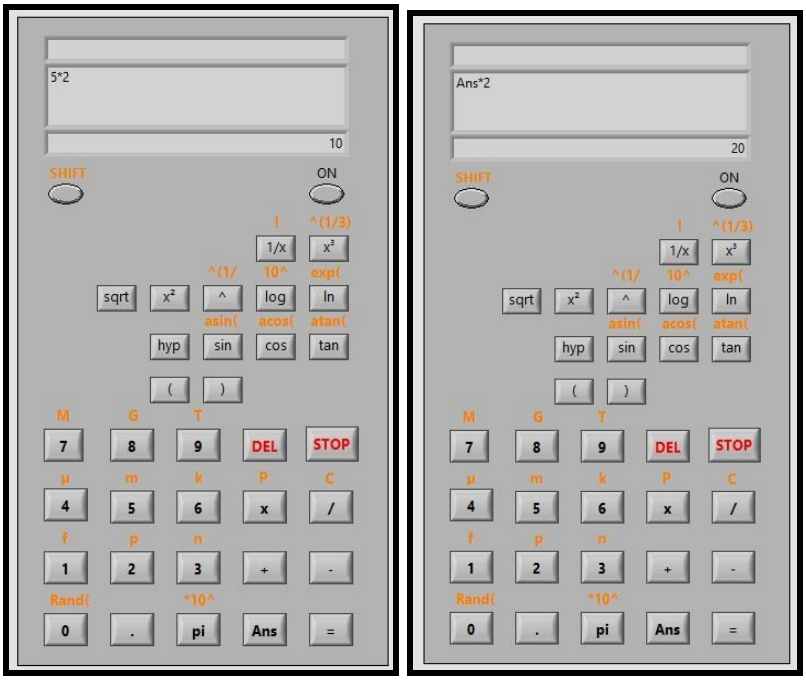
4. Complex Number Calculations
5. Statistical Calculations
6. Base-n Calculations
7. Equation Calculations

III. Features of Virtual Calculator

1. Basic Calculations:
    a. Using Engineering Symbols



2. Memory Functions:
    a. Answer Memory (Ans)



3. Function Calculations:
    a. $sin,\ cos,\ tan,\ sin^{-1},\ cos^{-1},\ tan^{-1},\ sinh,\ cosh,\ tanh,\ sinh^{-1},\ cosh^{-1},\ tanh^{-1}$

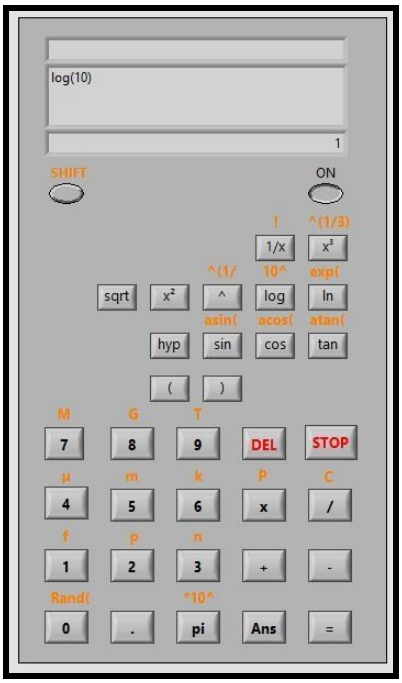

sin(1)
0.841470984807897
SHIFT
ON
! ^(1/3)
1/x x³
^(1/ 10^ exp(
sqrt x² ^ log ln
asin(acos(atan(
hyp sin cos tan
( )
M G T
7 8 9 DEL STOP
μ m k P C
4 5 6 x /
f p n
1 2 3 + -
Rand(*10^
0 . pi Ans =


asin(1)
1.570796326794897
SHIFT
ON
! ^(1/3)
1/x x³
^(1/ 10^ exp(
sqrt x² ^ log ln
asin(acos(atan(
hyp sin cos tan
( )
M G T
7 8 9 DEL STOP
μ m k P C
4 5 6 x /
f p n
1 2 3 + -
Rand(*10^
0 . pi Ans =


sinh(1)
1.175201193643801
SHIFT
ON
! ^(1/3)
1/x x³
^(1/ 10^ exp(
sqrt x² ^ log ln
asin(acos(atan(
hyp sin cos tan
( )
M G T
7 8 9 DEL STOP
μ m k P C
4 5 6 x /
f p n
1 2 3 + -
Rand(*10^
0 . pi Ans =


asinh(1)
0.881373587019543
SHIFT
ON
! ^(1/3)
1/x x³
^(1/ 10^ exp(
sqrt x² ^ log ln
asin(acos(atan(
hyp sin cos tan
( )
M G T
7 8 9 DEL STOP
μ m k P C
4 5 6 x /
f p n
1 2 3 + -
Rand(*10^
0 . pi Ans =

b. 10^x, e^x, log, ln

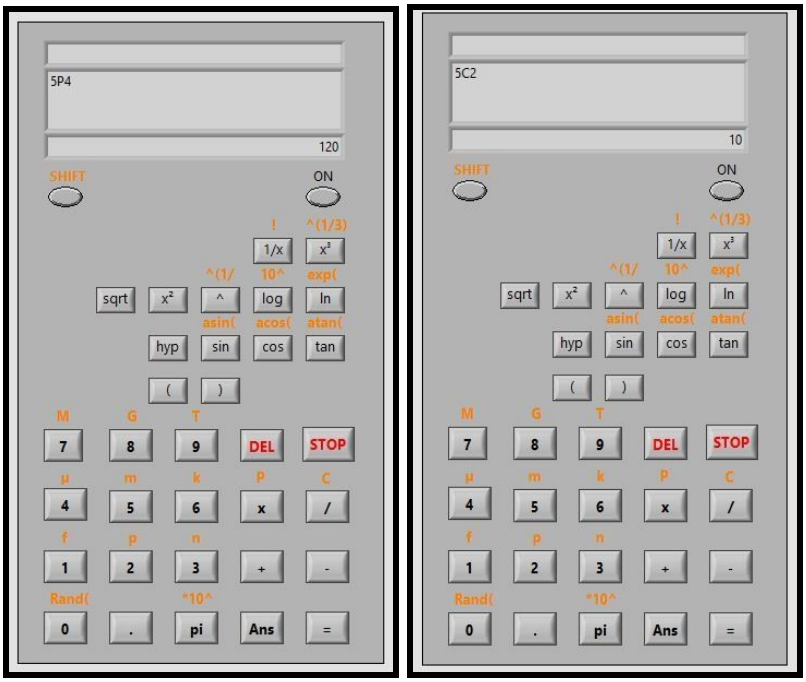


c. x^2, x^3, x^y, $\sqrt{\ }$, $\sqrt[3]{\ }$, $\sqrt[x]{\ }$, x^{-1}

d. $x!$

e. Random Number Generator

f. $nPr,\ nCr$



IV. Limitations of Virtual Calculator

1. The biggest limitation is the inability to control and display a pointer on the input screen. Such a pointer is vital for easy editing of expressions. But this is impossible on an indicator and hence we can only delete off operators and operands from the end of the expression. This also hinders editing the expression if a Syntax Error occurs - the workaround is to press SHIFT key and then DEL (character by character).
2. The methodology I applied was to use the `Label` and `Caption` of the boolean buttons to print the correct functionality on the input screen. But this limits us to only two functions per button.
3. Fractional display is not used in LabVIEW and hence it cannot be implemented easily.
4. Solving equations and integrals and derivatives are possible with certain blocks in LabVIEW, but then there is the added complexity of

the different modes the user may be in. Using an `Event Structure` inside another `Event Structure` may help this.

5. Another disadvantage is that the `Eval Formula String` VI only accepts and gives radian input and output and this is not easily changed.
6. Tracking previous expressions are possible with shift registers, but without a cursor this becomes pointless.