Abstract

Background/Objectives: The Engineering Unit Conversion (EUC) is essential for process monitoring and control in chemical or power plants. Digital data has to be converted back to natural form like temperature, pressure etc. Methods/Statistical Analysis: All received natural data and parameters are converted to digital domain for easy processing. This EUC does this operation to get back the values in the natural (physical) form. The complete C program for implementing EUC is developed using “Turbo C” compiler. It is validated with the known data and qualified for Temperature, Pressure & Flow parameters. Findings: For unit conversion complex methods are being used, it is to be eradicated through this application. Applications: This software is useful for process parameter monitoring and control in the nuclear power plants.

Keywords: Conversion, C Language, Digital Domain, Engineering Unit, Process Parameters

1. Introduction

In most of the power plants, the instrumentation and control plays an important role to get the information about the plant operation accurately and to act in the stipulated time. These systems are generally operated unmanned and continuous because of that it should be highly reliable and available. The operator views these parameter and initiate action based on the present values. In view of that it has to be converted to Engineering Unit so that the operator action is much more meaningful. In this context the Engineering Unit Conversion (EUC) becomes more important for precise and fast action of the operator, which in turn increases the productivity of the plant.

In any of the chemical plant or power generating plants, the automation of instrumentation and control becomes vital because the response to act for the events should be accurate, reliable and very fast. This requirement can be achieved by automating most of the safety actions since it has to operate on continuous basis without interruptions i.e. $24 \times 7$. Moreover, certain location it has to operate unmanned, in view of these requirements the process plants are normally instrumented with sensors and converted to digital values for further processing, control and safety actions and to display on the operator screen for ease in control and production.

2. Sensors

A sensor is a transducer whose purpose is to sense some characteristic of its environments. It detects events or changes in quantities and provides a corresponding output, generally as an electrical or optical signal$^1$. The different type of sensors is used in typical process plant to convert the physical parameters to electrical parameters which is generally in milli-volts or micro-volts.

The Table 1 shows the list of sensors used generally for measuring different parameters. This small voltage is amplified in two stages and they are called as pre-amplifier & post-amplifier to obtain better signal to noise ratio. The pre-amplifier is kept close to the sensor since noise level will be high. The pre-amplified the components are chosen
in such a way that they give good accuracy and very low drift because of temperature and aging. The high precision components were used. After two levels of amplification, the signal voltage is brought in the range of 4 to 20 ma current or 1 to 10Volts to have “Live Zero” concept.

3. Analog to Digital Converter

The analog value obtained from plant has to be converted to Digital to interface with the computer. To do this conversion Analog to Digital Converters (ADC) is used which converts on demand \(^2\). The output digital number represents the analog amplitude.

An Analog-to-Digital Converter (ADC) converts a continuous physical quantity (usually voltage) to a digital number that represents the quantity’s amplitude\(^2\). The conversion involves quantization of the input, so it necessarily introduces a small amount of error. Instead of doing a single conversion, an ADC often performs the conversions (“samples” the input) periodically. The result is a sequence of digital values that have been converted from a continuous-time and continuous-amplitude analog signal to a discrete-time and discrete-amplitude digital signal.

The amplified signal is taken to Analog to Digital Converter (ADC) and converted to digital counts of the range zero to 65536 (64K) using 16 bit ADC for data acquisition. The different types of ADC are available. The Successive Approximation type ADC used in this study is 16 bit resolution of bi-polar type, i.e., the input can be negative and positive voltage. The typical conversion time of this type of ADC is around 10 micro-second. This ADC requires Start of Conversion (SOC) to start the conversion and it produces End of Conversion (EOC) to indicate that conversion to digital for is complete and the data can be read from buffers.

4. Data Acquisition System

The Data acquisition is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer. Data Acquisition Systems (DAS) typically convert analog waveforms into digital values for processing\(^1\). The components of data acquisition systems include: Sensors, Signal conditioning circuitry and Analog-to-digital converters.

Data acquisition applications are usually controlled by software programs developed using various general purpose programming-languages such assembly, BASIC, C, etc. The Stand-alone data acquisition systems are often called data loggers.

There are also open-source software packages providing all the necessary tools to acquire data from different hardware equipment. These tools come from the scientific community where complex experiment requires fast, flexible and adaptable software.

5. Engineering Unit Conversion

Now the signal (parameter) value is available in digital count. To process further this has to be converted to engineering unit. This conversion we have achieved using C program which takes the count value and the type of sensor then it converts to the engineering value like Degree Centigrade, Cubic meter per hour, Kg per square cm etc.

| S.No. | Sensor                  | Physical parameter | Electrical output          |
|-------|-------------------------|--------------------|---------------------------|
| 1.    | Thermocouple, Resistance Temperature Detector (RTD), Thermistor | Temperature        | Microvolt to Millivolt    |
| 2.    | Venturimeter            | Pressure           | Millivolt                 |
| 3.    | Ultrasonic              | Level              | Time to counts            |
| 4.    | Eddy Current, Ultrasonic | Flow               | Millivolt                 |
| 5.    | Differential Pressure   | Density            | Volt                      |

**Table 1.** List of sensors

**Figure 1.** An approximation of \(f(x) = x^2\) at \((x, f(x))\).
Temperature Detector (RTD), Neutronic signals. Out of this RTD is a linear device so Equation 1 is followed. The Platinum 100 RTD is used which gives 100 ohms at room temperature and the resistance changes as the temperature changes. The change in temperature is sensed by passing constant current through the RTD and measuring the voltage across the “bridge circuit” so that errors are minimized and true value is obtained.

Output (RTD) = (Slope * ADC count) + Offset  

On the other hand K-type thermocouple in Non-linear and the conversion for this is done bye approximation at different points as shown in Figure 1 and tabulated as conversion table. The in between values of table is assumed as linear. The table size should be sufficiently large so that we get the final results without much of conversion error. Flow follows the square law as shown in Equation 2 and the neutronic signals follow the logarithmic equation.

Output (flow) = v^2 / (2g) + gz + (Pressure/density)  

where  
\( g \) is acceleration due to gravity.  

6. Pseudocode

The inputs for this EUC are Digital Count and Type of signal. The output is the engineering unit converted value with the corresponding units. The pseudo-code for implementing EUC is as follows:

1. Take the ADC Count
2. Check whether count it is within the range (0 to 64k)
3. Take the type of signal
4. Check whether type is within the types (0 to 3)
5. Switch to different sub-functions based on the type
6. Type Case is zero, do conversion for RTD
7. Type Case is One, do conversion for Thermocouple
8. Type Case is Two, do conversion for flow
9. Type Case is Three, do conversion for Neutronics
10. Print the converted value along with the corresponding units

7. Implementation

To implement the EUC the “C” language is chosen because of its simplicity and the availability of the compiler. The simple structured way the complete functionality is written using the C syntax. The complete program is compiled using Turbo C Compiler. The simple structures were used to implement this conversion. The EUC c program is converted into executable under MS-DOS; it created the file as EUC.EXE with the necessary object files binding. The EXE file can execute independently without any other program and it can be distributed for its use anywhere. The size of the EXE file is around 600 Kilobytes.

8. Testing

The compiled file EUC.EXE is run on the MS-DOS command prompt as shown in Figure 2. The ADC count of random values from zero to 64k with different types of conversion fed in to the program and the result were recorded. It is checked against manual calculation and verified all the results. All the results were matching and thus the program is validated.

Figure 2 shows the calculation performed with ADC count of 32000 and the type is 1. It means it is for K-Type thermocouple with the range of 0 to 1200 degree centigrade. The calculation goes like this. The full ADC count 65536 corresponds to 1200 degree centigrade. Already for this conversion a translation table for linearization is stored as array in C program. This translation table is stored for each 50 counts of ADC. So, First array index is calculated by dividing ADC count with 50. In this case we get the index as 640. The maximum value of array is 1310. The 640 element and the next elements are taken and corrected for the residual value and then result is calculated as 488.28. This value is printed on the screen as 488.28 °C.

In the same way, different values were fed as “test cases” using boundary Value analysis to cover the complete range of valid values. Similar test cases were generated for the other type of sensors and validated the software for its accuracy and response.

The proper comments were added to improve the readability and maintainability of the software. In the same
way the quality parameters like cyclomatic complexity, nesting and other values were kept as low as possible to satisfy the quality aspect of the software. 

9. Future Work

The software is written in C language to cover for different type of sensors. In future it can be extended to cater to many more type of sensors.

10. Conclusion

The Engineering Unit Conversion software is successfully completed using “C” as the programming language on the Microsoft Windows platform for four different sensors. It has the provision to include some more conversions which may emerge in due course of time.

11. Acknowledgement

The author sincerely thanks officials of Indira Gandhi Centre for Atomic Research, Kalpakkam, Department of Computer Science – Vel Tech High Tech Dr. Rangajran Dr. Sakunthala Engineering College and Department of Information technology – Vel Tech High Tech Dr. Rangajran Dr. Sakunthala Engineering College, for encouraging me to carry out this work and for their immense help in completing this work in time.

12. References

1. Sensor. Available from: https://en.wikipedia.org/wiki/Sensor, Date accessed: 21/8/2015.
2. Lathi BP. Modern Digital and Analog Communication Systems, 3rd edition. Oxford University Press, Inc.: New York, USA, 1998. p. 50–5.
3. Young SS. Computerized Data Acquisition and Analysis for the Life Sciences. Cambridge University Press: UK, 2001.
4. Thermocouples: Simple but misunderstood. Available from: http://www.edn.com/design/test-and-measurement/4423409/Simple--but-misunderstood, Date accessed: 21/8/2015
5. Linearization. Available from: https://en.wikipedia.org/wiki/Linearization, Date accessed: 21/8/2015
6. Bernoulli’s Principle. Available from: https://en.wikipedia.org/wiki/Bernoulli’s_principle, Date accessed: 21/8/2015
7. Kanetkar YP. Let us C, 6th Edition, Jones and Bartlett Publishers Inc: USA, 2008.
8. Turbo-C-Compiler, Available from: http://www.anzaq.com/2013/01/download-turbo-compiler-for.html, Date accessed: 21/8/2015
9. Sholiyi A, Alzubi JA, Alzubi OA, Almomani O, O’Farrell T. Near Capacity Irregular Turbo Code. Indian Journal of Science and Technology. 2015 Sep; 8(23):1–10.
10. Pressman RS. Software Engineering: A Practitioner Approach, 6th edn., McGraw Hill: New York, 2005.
11. Sommerville I. Software Engineering, 9th Edition, Addition Wesley Pearson: USA, 2011.