



INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

RLE Compander for Wireless Sensor Data Processing

Arun Kumar.R

Assistant Professor, CSE, Vidya Jyothi Institute of Technology, Hyderabad

Abstract

RLE data compression is effective when the measured signals are expected to be slow and repeatable. The task of this unit is crucial to the system power performance. RLE is a conceptually simple form of compression. RLE consists of the process of searching for repeated runs of a single symbol in an input stream and replacing them by a single instance of the symbol and a run count. The aim of this paper is to model and design an efficient embedded wireless system that is easy to integrate with other technologies or infrastructures at a low cost. The system would read analogue information recorded by sensors (physical parameters) in a transmitting unit attached to the system. The recorded data are converted digitally using analogue-to-digital converter and sent to on/off frequency shift keying (FSK) transmitter through Micro controller interfaced to the system. System functions such as bus interfacing, data buffering, compression and data framing are implemented in Embedded C software. Remotely connected second Micro controller is interfaced to the output of the receiver and is responsible for processing the data inversely like de-framing and de-compressing. This system may be used as general-purpose wireless sensor data compressing and processing in many fields of interest such as mineral processing, patient monitoring, sensor data instrumentation, Bio medical instrumentation and radar sensor data processing etc. The data transmission and reception are implemented effectively with packet based communication protocol with data integrity. In this project we used the HDLC (High Data Link Controller) protocol, this unit is considered to be the main core of the system model, where the data are grouped into frames and sent to the transmitter. HDLC. This project implementation includes various hardware and software modules: Micro controller, LCD display, keyboard, encoder/decoder, RF modems and RPS design and embedded firmware modules to drive hardware modules.

Keywords: RLE, FSK, EMBEDDED C SOFTWARE, COMPRESSION, WIRELESS SENSOR, HDLC

Introduction

RLE data compression is effective when the measured signals are expected to be slow and repeatable. The task of this unit is crucial to the system power performance. RLE consists of the process of searching for repeated runs of a single symbol in an input stream, and replacing them by a single instance of the symbol and a run count.

The aim of this paper is to model and design an efficient embedded wireless system that is easy to integrate with other technologies or infrastructures at a low cost. The system would read analogue information recorded by sensors (physical parameters) in a transmitting unit attached to the system. The recorded data are converted digitally using analogue-to-digital converter and sent to on/off frequency shift keying (FSK) transmitter through Micro controller interfaced to the system.

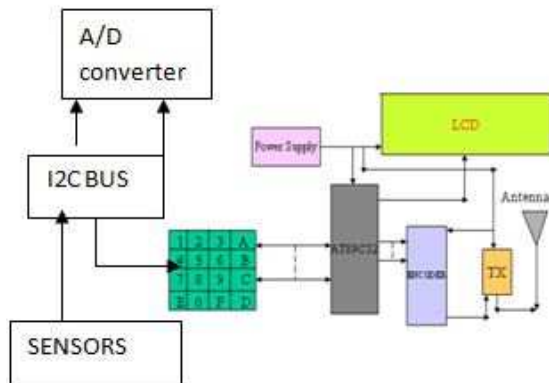
System functions such as bus interfacing, data buffering, compression and data framing are implemented in Embedded C software. Remotely connected second Micro controller is interfaced to the output of the receiver and is responsible for processing the data inversely like de-framing and de-compressing. This system may be used as general-purpose wireless sensor data compressing and processing in many fields of interest such as mineral processing, patient monitoring, sensor data instrumentation, Bio medical instrumentation and radar sensor data processing etc. The data transmission and reception are implemented effectively with packet based communication protocol with data integrity.

In this project we used the HDLC (High Data Link Controller) protocol, this unit is considered to be the main core of the system model, where the data are grouped into frames and sent to the transmitter. HDLC protocol is a bit-oriented protocol that is used as a data link for most of the current

communication systems. This paper implementation includes various hardware and software modules.

Block Diagram

Transmitter Section: At the transmitter end the required temperature sensor is interfaced to the microcontroller along with the a/d converter and I2C bus. The major reason of using the a/d converter is because the values recorded by the sensors are in analog nature and these analog values are to be converted to digital values. The reason for the use of a I2C bus is when we have a number of sensors to be interfaced for the application then this would be serving in the saving of pins of the microcontroller. We have the microcontroller programmed in two modes of demonstrating the functionality of this RLE algorithm. The first mode being the data entered from the keyboard and the second one being the values recorded by the sensor values. At the transmitting side we have the keyboard interfaced to port 1 and the lcd interfaced to port 0 of the microcontroller. In the first mode we assume the data to be entered data from the keyboard a sample of ten values is assumed and entered. And in the second values from sensor are taken. At the transmitting end the values are converted into digital equivalents if the input mode is from sensor or the data from keyboard is fed to the microcontroller and is displayed on the lcd interfaced on the transmitting end. The data is converted into serial data using the encoder interfaced to the AT89C52. The TE pin of the encoder is interfaced to pin 2 of port 3 of AT89C52. And the data output from the encoder is fed back to the RF transmitter which uses on off shift keying technique.



Receiver Section: At the receiver end we have the decoder and an lcd interfaced to the AT89C52 and a RF receiver module. At this end the transmitted signal count and the data symbol which is repeated is received through the receiver module and is decompressed. The data and signal which are received are converted into parallel data using the decoder interfaced at the receiver side to the

AT89C52 microcontroller. At the receiver side the decoder's pin VT is interfaced to pin 2 of port 3 of the microcontroller. The serial data is converted to parallel data from the decoder and then is displayed on the lcd interfaced to port 0 of the microcontroller. All the functionalities are implemented through the use of microcontroller and through programming it by the use of embedded C language. Thus the functionality of compression of data and decompression of repeated symbols is achieved.

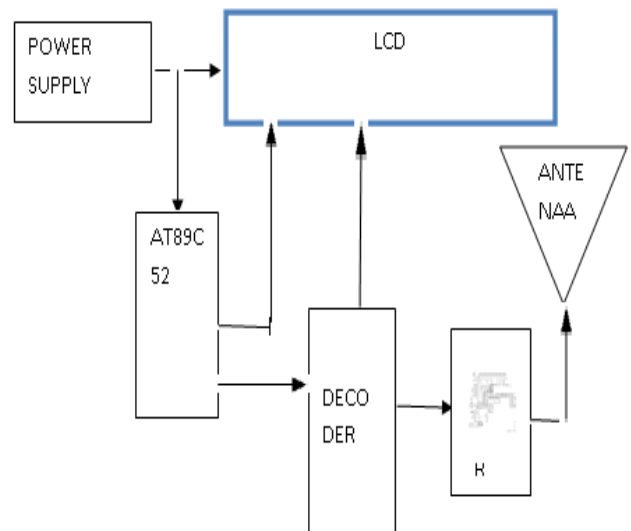


Figure 1.2: RECEIVER SECTION

Transmitter Schematic Diagram

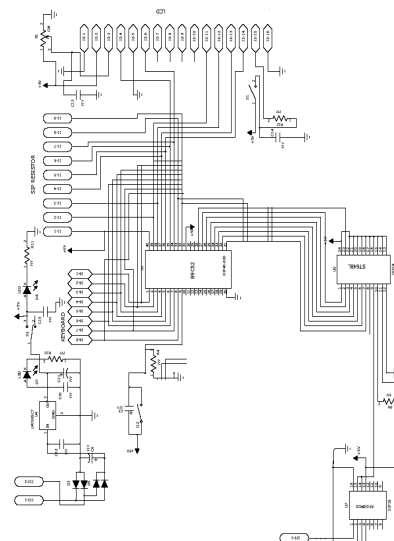


Figure 1.3

Receiver Schematic Diagram

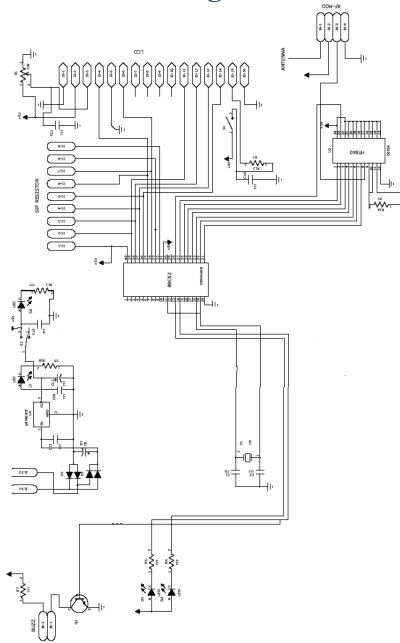


Figure 1.4

Schematics Explanation

The above two diagrams show the schematics of the transmitter and the receiver. At the transmitter the keyboard is interfaced to the port 1 of the microcontroller. An lcd is interfaced to port three. The sensor is interfaced at the at the port 1 it self. It has two modes of operation one when the input is taken from sensor and one when taken from the keyboard .the data is fed to the microcontroller and through the encoder the data is converted to serial form whose pin TE is interfaced to P3.2 and is transmitted through the use of RF transmitter module. When many sensors are needed to be interfaced to the required applications then we use the I2C bus which facilitates the interfacing of many required sensors and also enables the savage of pins of the microcontroller.

At the receiving end the RF receiver module receives the count and the data which is repeated. The data received is sent through the decoder for parallel conversion and fed through the microcontroller. The decoder pin VT (valid transmission) is connected to P3.2 at the receiving end. The program is prewritten in embedded C language such that it expands according the count and the symbol. At this receiving end we interface an lcd also to port 0 of the microcontroller which facilitates the visualization of decompression phenomenon being done.

Compression is useful because it helps reduce resource usage, such as data storage space or transmission capacity. Because compressed data must be decompressed to use, this extra processing imposes computational or other costs through decompression; this situation is far from being a free lunch. Data compression is subject to a space–time complexity trade-off. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in full before watching it may be inconvenient or require additional storage. The design of data compression schemes involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (e.g., when using lossy data compression), and the computational resources required to compress and uncompress the data.^[5]

New alternatives to traditional systems (which sample at full resolution, then compress) provide efficient resource usage based on principles of compressed sensing. Compressed sensing techniques circumvent the need for data compression by sampling off on a cleverly selected basis.

Flow Chart at Transmitter

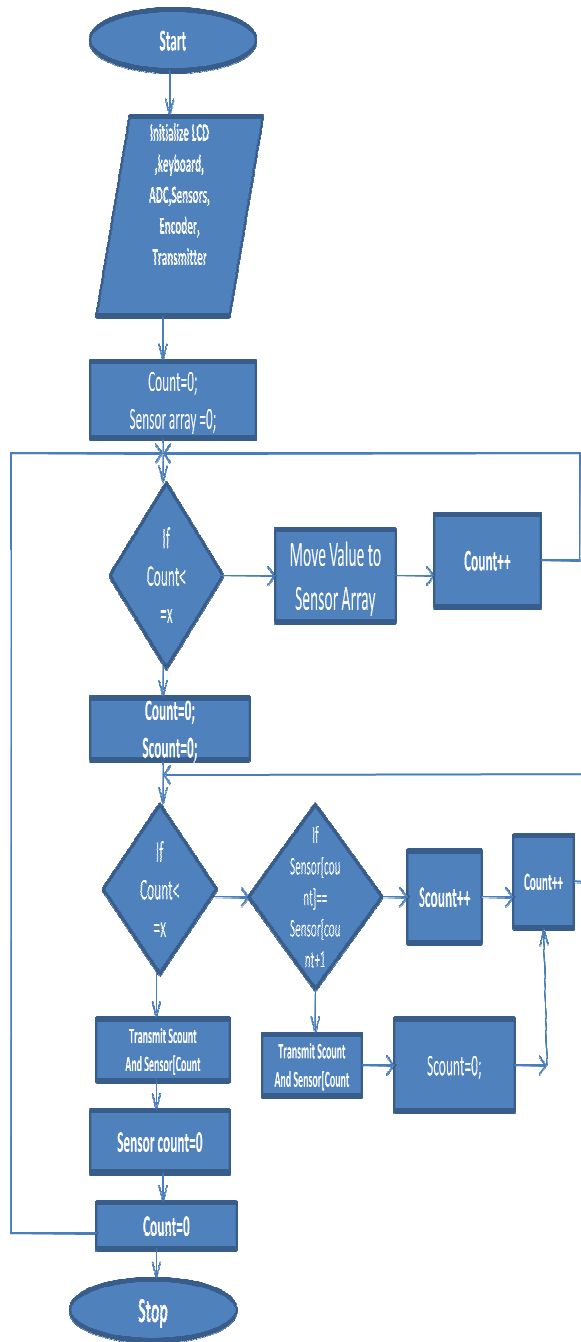


Figure 1.5

Flow Chart At Receiver

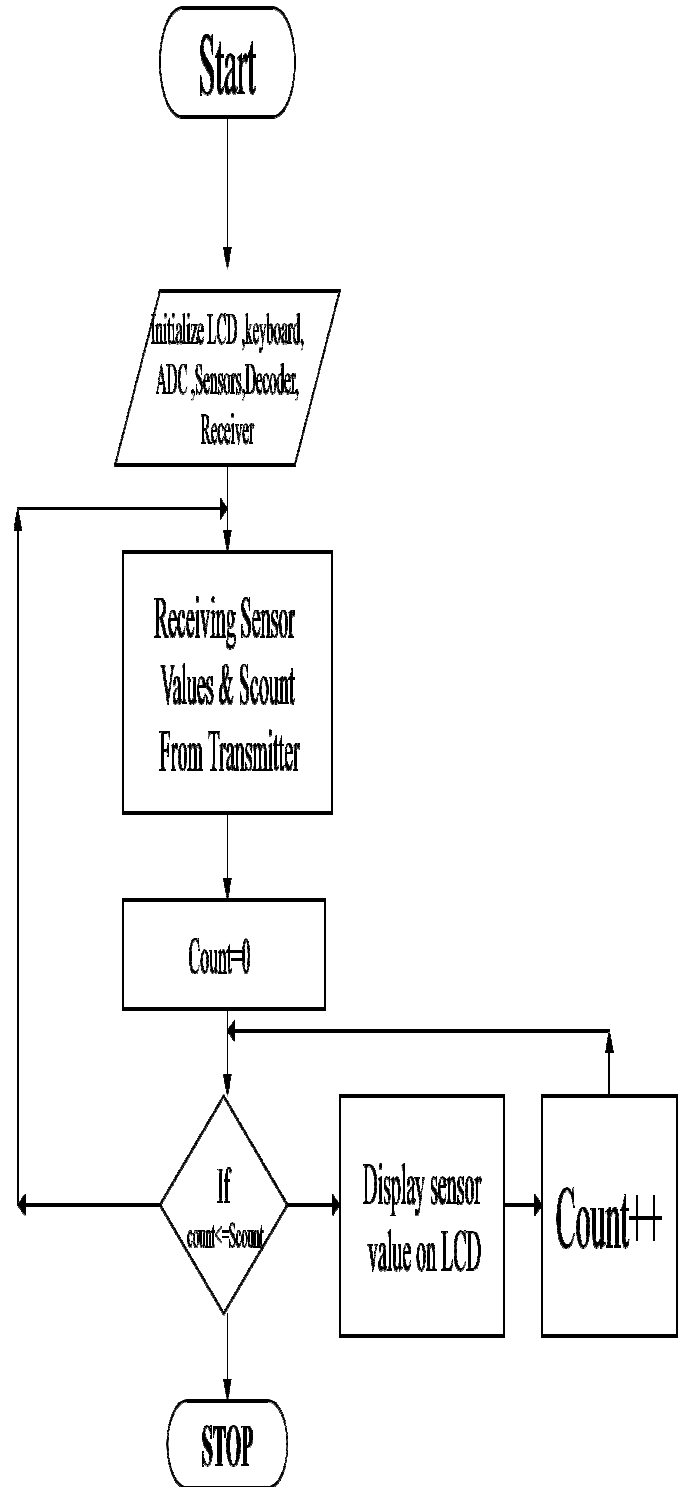


Figure 1.6

Results

1. Click on the Keil uVision Icon on Desktop
2. The following fig will appear

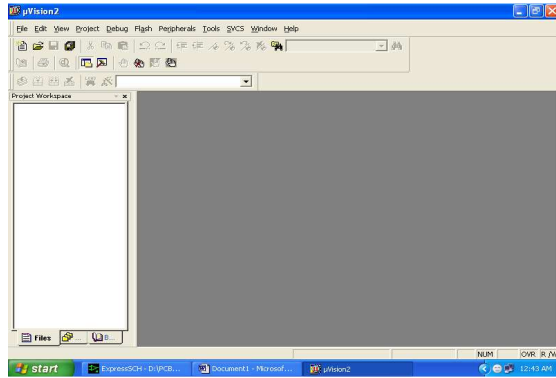


Fig 7.1

3. Click on the Project menu from the title bar
4. Then Click on New Project

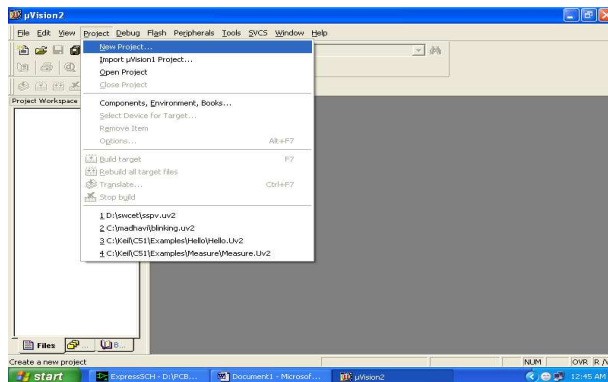
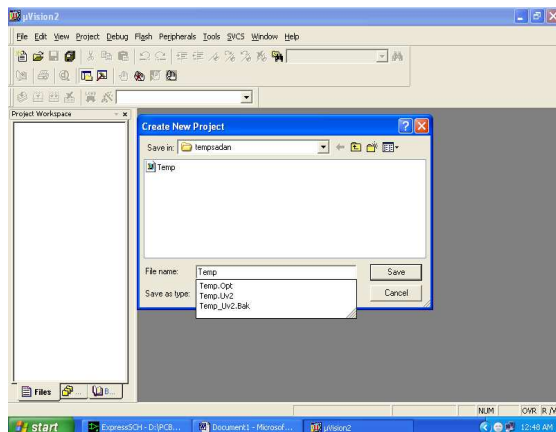


Fig 7.2

5. Save the Project by typing suitable project name with no extension in your own folder sited in either C:\ or D:\



14. Now double click on the Target1, you would get another option “Source group 1” as shown in next page.

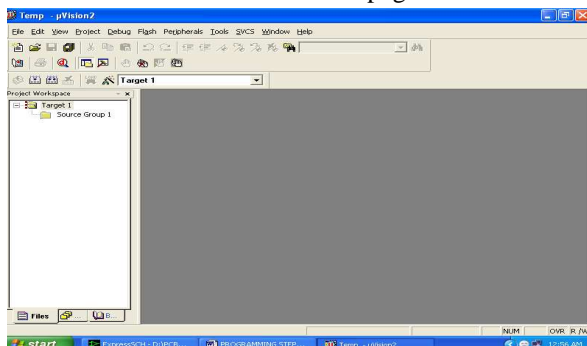


Fig 7.6

15. Click on the file option from menu bar and select “new”

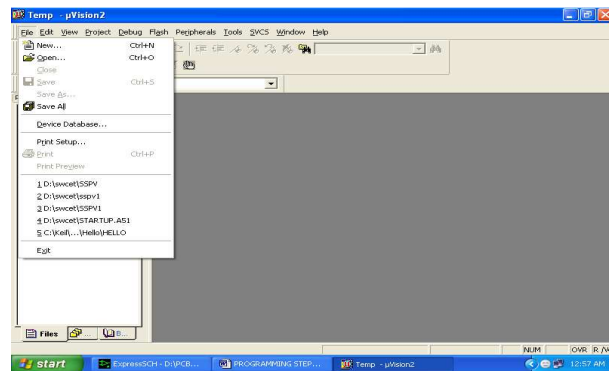


Fig 7.7

16. The next screen will be as shown in next page, and just maximize it by double clicking on its blue boarder.

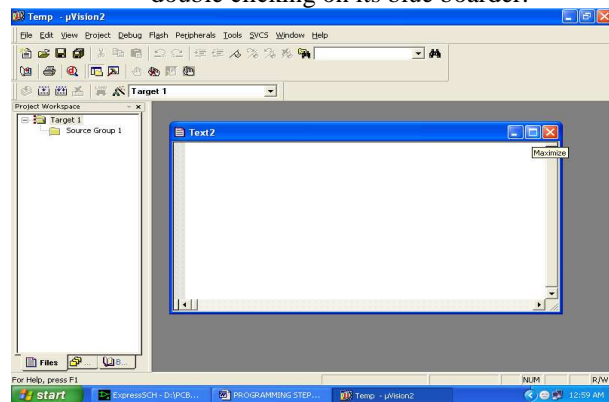


Fig 7.8

17. Now start writing program in either in “C” or “ASM”
18. For a program written in Assembly, then save it with extension “.asm” and for “C” based program save it with extension “.C”

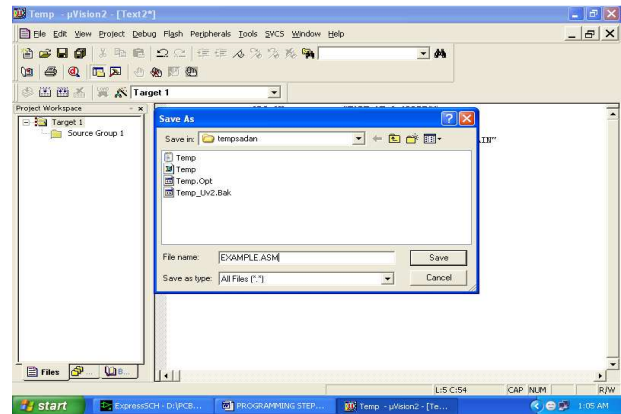


Fig 7.9

19. Now right click on Source group 1 and click on “Add files to Group Source”

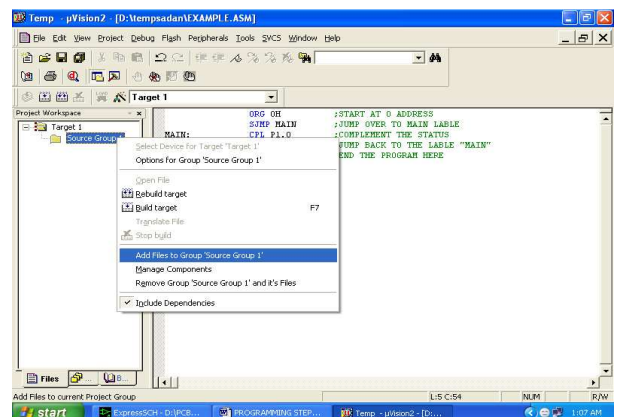


Fig 7.10

20. Now you will get another window, on which by default “C” files will appear.

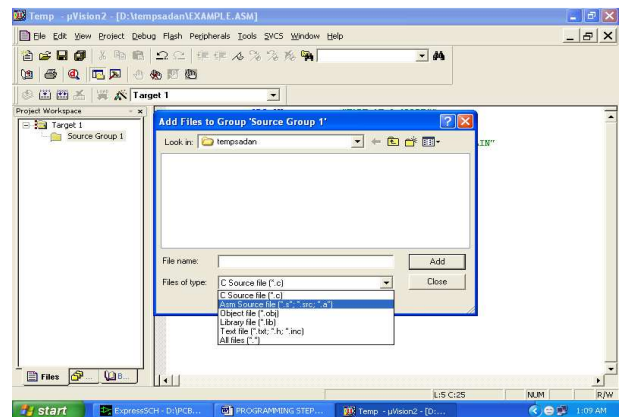


Fig 7.11

21. Now select as per your file extension given while saving the file
22. Click only one time on option “ADD”

23. Now Press function key F7 to compile. Any error will appear if so happen.

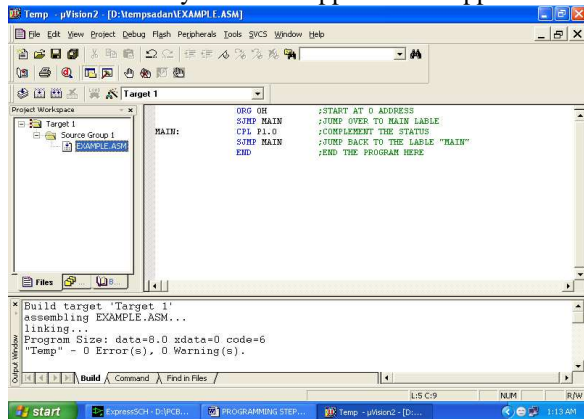


Fig 7.12

24. If the file contains no error, then press Control+F5 simultaneously.
25. The new window is as follows

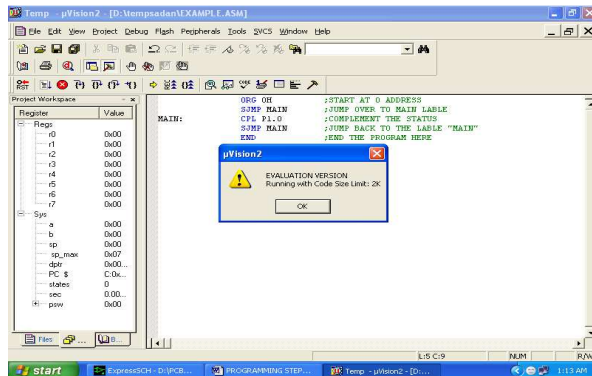


Fig 7.13

26. Then Click “OK”
27. Now Click on the Peripherals from menu bar, and check your required port as shown in fig below

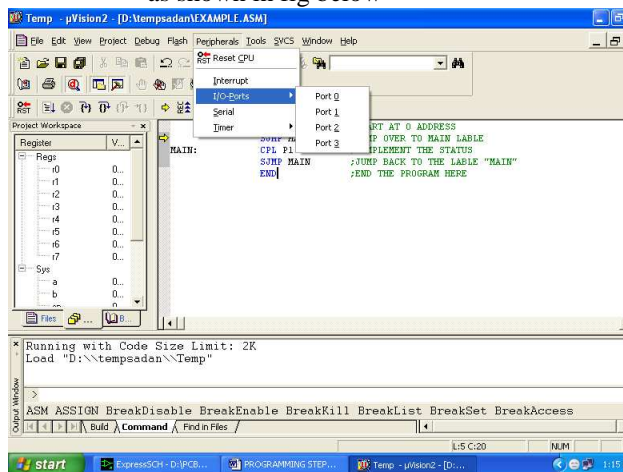


Fig 7.14

28. Drag the port a side and click in the program file.

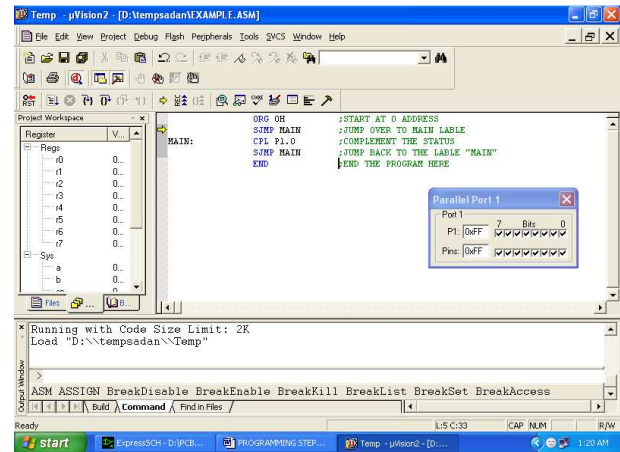


Fig 7.15

29. Now keep Pressing function key “F11” slowly and observe.
30. You are running your program successfully

Conclusion

The paper “RLE COMPANDER FOR WIRELESS SENSOR DATA PROCESSING” has been successfully designed and tested. Integrating features of all the hardware components used have been reasoned out and placed carefully thus contributing to the best working of the unit. Secondly, using highly advanced IC’s and with the help of growing technology the paper has been successfully implemented.

References

- [1] L. Liu, X. Zhang, And H. D. Ma, “Dynamic Node Collaboration For Mobile Target Tracking In Wireless Camera Sensor Networks,” In *Proc.Ieee Infocom 2009, Rio De Janeiro, Brazil, Apr. 19–25, 2009, Pp.1188–1196*
- [2] Akkaya, K., Younis, M.: *A Survey On Routing Protocols For*
- [3] *Wireless Sensor Networks.Ad Hoc Networks* 3(3), 325–349 (2005)
- [4] Akkaya, K., Younis, M.: *A Survey On Routing Protocols For Wireless Sensor Networks.Ad Hoc Networks* 3(3), 325–349 (2005)
- [5] Akkaya, K., Younis, M.: *A Survey On Routing Protocols For Wireless Sensor Networks.Ad Hoc Networks* 3(3), 325–349

- (2005)8051-Microcontroller And Embedded System.- Mohd. Mazidi. , Janice Gillispie Mazadi
- [6] A. Klein And Y. Tsividis. Instantaneously Companding Digital Signal Processors. *Ieee Icassp*, 3:1433–1436, April 2007.
- [7] L. Toth Y. Tsividis, V. Gopinathan. Companding In Signal Processing. *Electronicletters*, 26:1331–1332, August 1990.
- [8] N. Krishnapura Y. Tsividis. Internally Varying Analog Circuits Minimize Powerdissipation. *Circuits And Devices Magazine, Ieee*, 19:63–72, January 2003.
- [9] T. Linder, “On asymptotically optimal companding quantization,” *Prob. Contr. Inf. Theory*, vol. 20, no. 6, pp. 475–484, 1991.