

## Documentation for Compress (compress\_29Dec.cpp)

*Overview:* Take real data files in binary after Level 2 and compress each packet in the file individually, adding the information of the minimums and ranges for each channel, then write all new compressed packets to another .bin file.

*Method:* Handles multiple run files, looping through node and slot number (i.e. run15703\_node\*\_slot\*\_uncompressed.bin), and reads 1038 words at a time (number of words in an uncompressed packet - 2 ADC words, 6 header words, 1024 energy words, 6 COE words). The packet is put into a function that flips the endianness of the words in the packet. The next function finds the minimum and maximum for each channel, noting that the packets are arranged by sample number, with all 16 channels for sample 1 listed first, then all 16 channels for sample 2 next. It uses the minimum and maximum within each channel to calculate the range of bits needed to express the difference between the minimum and maximum, storing the minimum and range between the original 8 header and ADC words in the packet and the new compressed words (words 9-40). The next function executes the compression, taking the original energy word, subtracting the minimum from its corresponding channel, and using the binary range to represent the remainder. All 16 channels are then written using their ranges into a new set of words (see compression example below, flowing from left to right). Lastly, the endianness of the words are flipped back.

Example for compressing one sample by hand, after the minimum and range are found:

Uncompressed Data						Compressed Output	
Channel	Minimum	Range	Energy word	Difference (EWord - Min)	Binary (Diff written in Binary Range)	Binary	Hex
0	0	14	100	100	00000001100100		
1	589	4	591	2	0010	1000 0000 0110 0100	8064
2	545	4	546	1	0001	1011 0010 0001 0010	B212
3	556	4	558	2	0010	1000 1100 0100 0100	8C44
4	550	4	553	3	0011	1011 0011 0001 0011	B313
5	527	4	528	1	0001	1001 1010 0001 0000	9A10
6	543	4	545	2	0010	1000 0000 0011 1111	803F
7	508	4	511	3	0011		
8	527	4	530	3	0011		
9	570	4	571	1	0001		
10	501	4	504	3	0011		
11	535	4	538	3	0011		
12	552	4	556	4	0100		
13	574	4	582	8	1000		
14	582	4	588	6	0110		
15	0	14	63	63	0000000111111		

*More detail by function:*

- In main:
  - Finds the size of the input file and uses that to determine the number of packets that will be compressed
    - Each packet comprised of 1038 words (2 bytes for each word)
  - Takes in 1038 words at a time to apply the compression on
  - Prints the compressed packet to a .bin file
- “FlipStuff” function: endianness is changed from its input as big endian into little endian
- “FindMinRange” function: The minimum and range for each channel are determined
  - Goes through the 64 samples for each channel and stores the minimum and maximum energy word
  - The range needed to store the difference is calculated (max - min)
  - The number of bits necessary for the difference is stored as the range for each channel
- “Compress” function:
  - Difference from the minimum per channel to the energy word is calculated for each sample and stored in binary
  - The bit range is used to mask and place the binary difference for each channel together (reference the corresponding color coded Binary column--left--to the compressed binary words on the right) to make the compressed words (labeled as Hex in the reference figure)
  - Note: The “1 0” included at the beginning of each compressed binary word is a label included on the two most significant bits to indicate that the words are energy words
  - The compressed array is then fed back to the main function, where it is printed (amended to the end) of a .bin file

*Results:* A compressed file that contains the same number of packets as the uncompressed file it reads from. Each packet after compression has 6 headers, 2 ADC words, 16 minima, 16 ranges, \*\* compressed energy words, and 6 COE words. The compressed energy words are a variable number, depending on the range for the 16 channels, but should be divisible by 64 due to the 64 samples. This is stored in a .bin file with the same name as the original except that “uncompress” is now “compressed” (i.e. run15703\_node\*\_slot\*\_compressed.bin).