

A Deeper Look into Open Wire Detection on Battery Management Systems

Frank Zhang, Applications Engineer

Abstract

The open wire detection function plays a crucial role in the safe and reliable operation of the battery management systems (BMS). Due to its critical importance, it is recommended for those interested in BMS or involved in BMS design to take the time to understand this function. Taking Analog Devices' cell monitor as an example, this article discusses in detail how BMS circuits can accurately identify almost all instances of open wire connections when interfacing with external cells using algorithms. The discussion on open wire detection algorithms aims to provide readers with a deeper understanding of this function within BMS. The pseudocode provided in this article for open wire detection is intended to serve as a design reference for BMS designers.

Introduction

In the battery management system (BMS), there are extensive wiring connections between individual cells and cell monitor circuits. These wiring connections are essential for enabling reliable monitoring of cell parameters, including voltage, current, temperature, and others by the cell monitors. Additionally, these wiring connections may serve as the current path for cell passive balancing discharge or the transmission path for relay control signals. The number of cells to be managed in BMS is typically very large, necessitating the use of a significant amount of wiring connections. These wiring connections are numerous, some even lengthy, as they often need to span between different printed circuit boards (PCBs) and between PCBs and battery packs (comprising numerous individual cells). They also require many connecting components to be used in conjunction. The occurrence of open wires in the BMS is to be vigorously avoided. After all, if a cell experiences open wires, it means that effective monitoring of its state will be weakened or lost, and unmonitored cells will pose a hidden safety hazard to the entire BMS, with the threat of becoming fatal to the system at any unknown time. When open wires do occur, the primary task is to quickly, accurately, and efficiently determine the location of the open wires and provide timely notifications. An effective and accurate open wire detection algorithm will greatly enhance the reliability of the BMS and facilitate troubleshooting for both the BMS and battery pack. Accurate fault localization by the algorithm can effectively reduce many unnecessary repetitive checks, as well as disassembly and assembly work, during manual troubleshooting processes.

Open Wire Detection in the C Pin of a BMS

Due to the critical importance of open wire detection in a BMS, most of ADI's cell monitors are equipped with open wire detection commands and their respective implementation methods. The open wire detection methods used in different models of ADI cell monitors may vary, such as those used in the [ADBMS6830B](#) differing from those used in the [LTC6813](#) (the latter of which is the focus of this article). The open wire detection method employed by the LTC6813 is also a common method used in various models of ADI cell monitors, possesses generality, and is based on the ADOW (open wire check) command.

The ADOW command is used to check for any open wires between the ADCs of the [LTC6813-1](#) and the external cells. This command performs analog-to-digital conversions on the C pin inputs identically to the ADCV (measuring cell voltages) command, except two internal current sources sink or source current into the two C pins while they are being measured. The pull-up (PUP) bit of the ADOW command determines whether the current sources are sinking or sourcing 100 μ A. The following simple algorithm can be used to check for an open wire on any of the 19 C pins:

- ▶ Run the 18-cell command ADOW with PUP = 1 at least twice. Read the cell voltages for cells 1 through 18 once at the end and store them in array CELLPU(n).
- ▶ Run the 18-cell command ADOW with PUP = 0 at least twice. Read the cell voltages for cells 1 through 18 once at the end and store them in array CELLPD(n).
- ▶ Take the difference between the pull-up and pull-down measurements made in the previous steps for cells 2 to 18: $CELL\Delta(n) = CELLPU(n) - CELLPD(n)$.
- ▶ For all values of n from 1 to 17: If $CELL\Delta(n+1) < -400\text{mV}$, then C(n) is open. If $CELLPU(1) = 0.0000$, then C0 is open. If $CELLPD(18) = 0.0000$, then C18 is open.

The description in the data sheet provides a clear and understandable explanation of the ADOW open wire detection command and its algorithm implementation used in the LTC6813. Figure 1 shows the algorithm in a graphical form to help readers visualize the intricacies of the algorithm more intuitively.

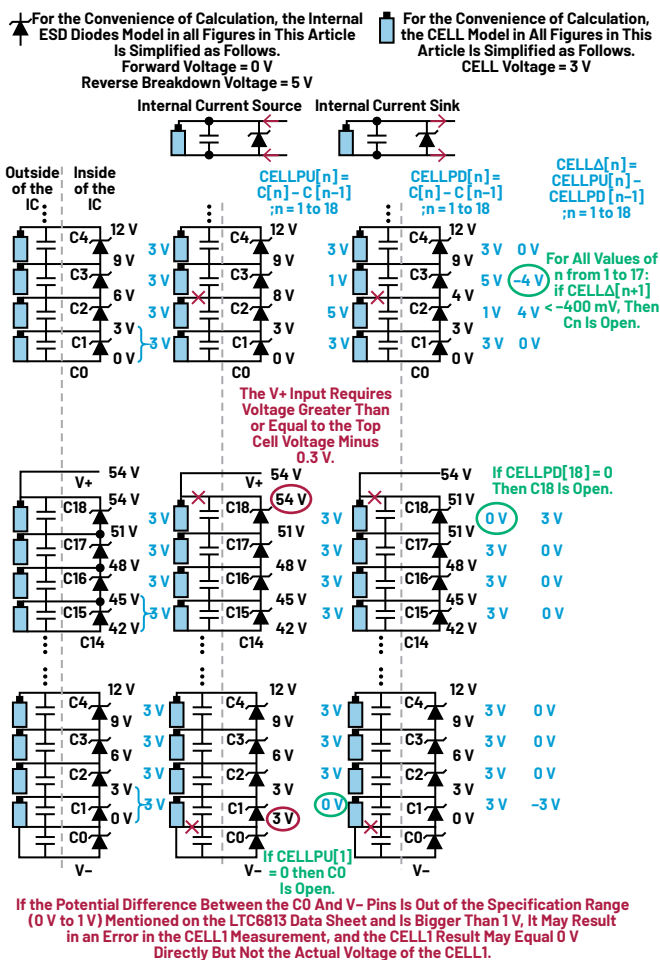


Figure 1. Algorithm diagram for open wire detection in the LTC6813.

In addition to the graphical demonstration of open wire detection based on the ADOV command shown in Figure 1, practical open wire detection experiments and waveform tests were conducted using an [LTC6813 evaluation board](#) and 18650 batteries. This allowed for a clearer understanding of the behavior of the circuit during open wire detection. The cells used in the experiment had voltages of around 4 V per cell, and the four probes of the oscilloscope were respectively connected to the C1/C2/C3/C4 pins of the LTC6813. During the experiment, voltage waveforms of CELL2/CELL3/CELL4 were measured separately. By combining the timing specifications for executing the ADCV command obtained from the data sheet, as shown in Figure 2 and Table 1, a better understanding of the waveform

detection results of this experiment was achieved. The timing specifications for the ADOV command execution can be referred to those of the ADCV command, as the ADOV command performs analog-to-digital conversions on the C pin inputs identically to the ADCV command, except two internal current sources sink or source current into the two C pins while they are being measured. The test results of the open wire detection experiment are shown in Figure 3 and Figure 4. In the circuit test of Figure 3, no open wire occurred, mainly serving as a reference experiment. In the circuit test of Figure 4, an open wire occurred at the C2 pin position, and the waveform test results of CELL2/CELL3/CELL4 under this open wire condition during the execution of the ADOV command were provided. From the test results, the observed changes in circuit behavior were consistent with the depiction in Figure 1.

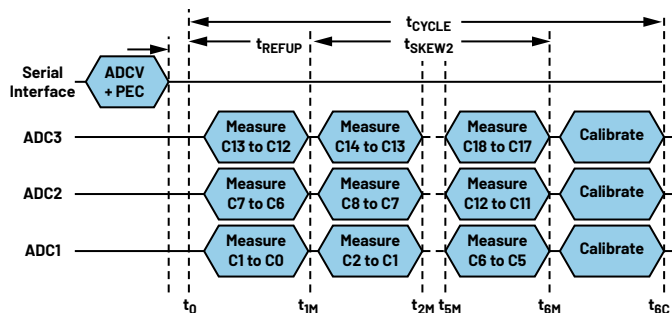


Figure 2. Timing for ADCV command measuring all 18 cells.

Table 1. Conversion and Synchronization Times for ADCV Command Measuring All 18 Cells in Different Modes

Mode	Conversion Times (μs)						Synchronization Time (μs)
	t ₀	t _{1M}	t _{2M}	t _{5M}	t _{6M}	t _{6C}	
27 kHz	0	58	104	244	291	1,121	233
14 kHz	0	87	163	390	566	1,296	379
7 kHz	0	145	279	681	815	2,343	670
3 kHz	0	261	512	1,263	1,413	3,041	1,252
2 kHz	0	496	977	2,426	2,909	4,437	2,415
1 kHz	0	960	1,908	4,753	5,702	7,230	4,742
422 Hz	0	1,890	3,777	9,408	11,287	12,816	9,397
26 Hz	0	29,818	59,624	149,044	178,851	201,325	149,033

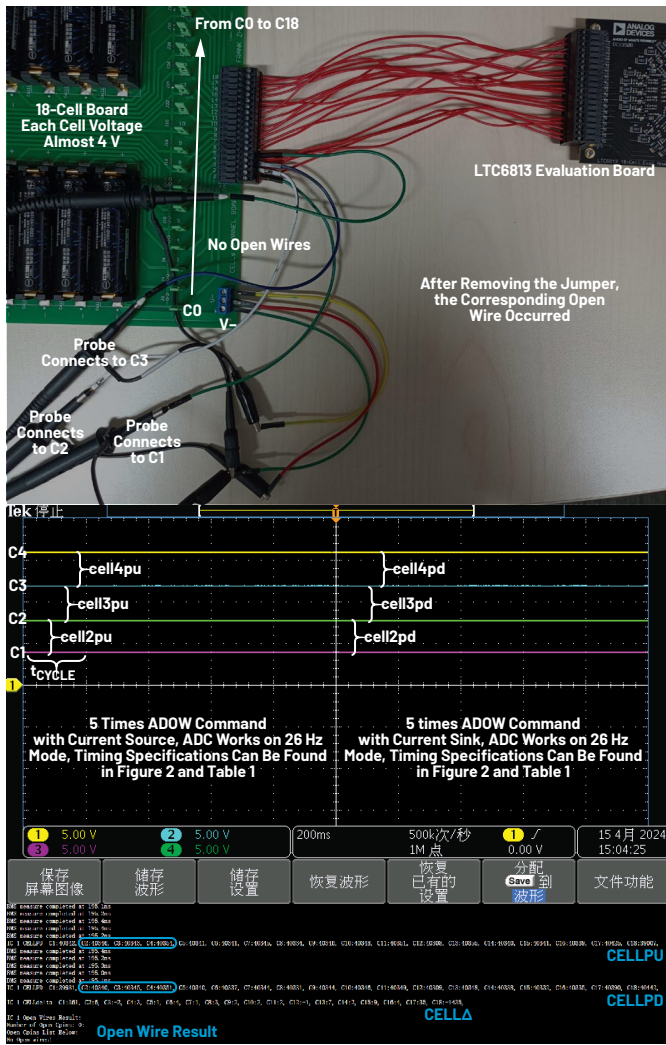


Figure 3. The experimental results of open wire detection and waveform testing (Experiment 1).

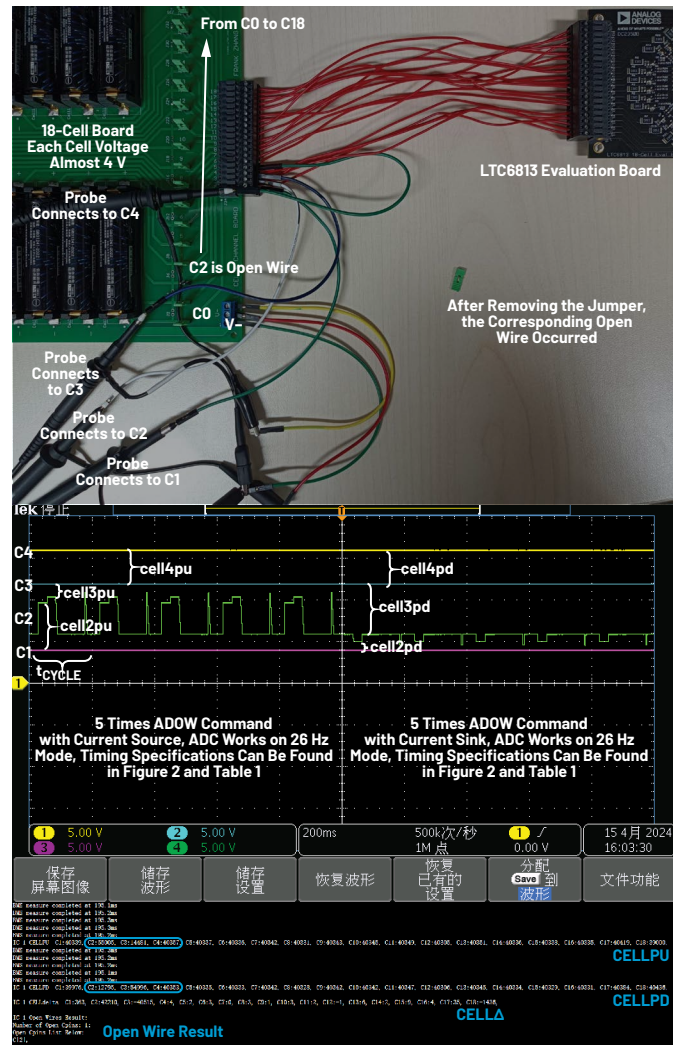


Figure 4. The experimental results of open wire detection and waveform testing (Experiment 2).

Issues When Multiple Open Wires Occur in the BMS

Imagine facing a serious scenario in the BMS where multiple wires are open, such as more than one C pin in a cell monitor circuit. While this situation is unlikely, its low probability doesn't make it impossible. In such a situation, the algorithm described in the data sheet is more effective for detecting open wires in one individual C pin. When it comes to multiple C pins experiencing open wires simultaneously, mechanically applying the algorithm from the data sheet may not yield accurate results.

For instance, applying the algorithm described in the LTC6813 data sheet to detect open wires in the scenario where multiple open wires occur as depicted in Figure 5, the result obtained by executing the algorithm from the data sheet is that only the open wires at the C6, C7, C8, and C9 pins can be detected, while the open wires at the C2, C3, C4, and C5 pins cannot be detected by the algorithm. This is not the desired result.

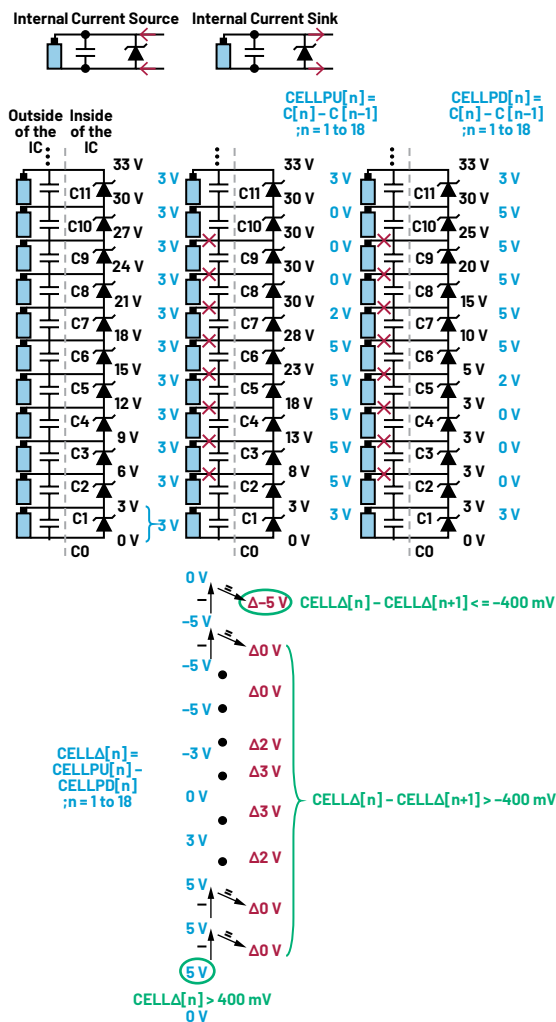


Figure 5. Detection algorithm diagram for multiple open wires occurring in the BMS (Example 1).

Nevertheless, the open wire detection algorithm mentioned in the data sheet remains highly effective in accurately identifying open wire in one individual C pin, and it also provides a valuable reference for detecting multiple open wire

situations. When an open wire condition arises in the C pin of the cell monitor, whether it's a single instance or multiple open wires, applying the algorithm outlined in the data sheet mechanically might not accurately detect all occurrences and their respective positions, especially when multiple open wires happen simultaneously. However, the algorithm will certainly alert the BMS to the presence of an open wire fault. If the BMS circuit board experiences an open wire error, the circuit user will need to manually inspect and repair the circuit more precisely, and typically, open wire errors that were not detected by the algorithm will be discovered during further careful manual inspection.

Algorithm for Detecting Multiple Open Wires on C Pins

Is there a better algorithm available that can rapidly, accurately, and efficiently identify all open wire situations within the cell monitor, even when multiple open wires occur simultaneously on C pins, without solely relying on manual inspection by experienced engineers to detect all open wire situations in the circuit? The answer is yes.

If we further observe the ADOW open wire detection algorithm and the data obtained in the data sheet, we can find that with just one additional calculation step, almost all open wire situations at C pins in the circuit can be accurately identified. This is true whether it is a single open wire or multiple open wires, and regardless of the location or number of open wires. Almost all, because in practical testing, it is still very difficult to accurately detect all open wire faults in the circuit when the C0 pin or C18 pin experiences an open wire situation. The open wire situations related to the C0 pin and the C18 pin will be discussed in a later part of this article. For now, let's focus on how to detect all open wire situations when pins C1 through C17 experience open wires.

The statement "just one additional calculation step" refers to simply calculating the difference between $CELL\Delta[n]$ and $CELL\Delta[n+1]$, which generates a highly efficient and accurate open wire detection algorithm. The implementation and steps of this algorithm are described below.

Step 1: First, following the approach described in the data sheet, send the ADOW command to the cell monitor to initiate open wire detection. Then, two internal current sources sink or source current into the two C pins while they are being measured. The voltage data for each cell is obtained through the ADCs in the chip and stored in three arrays: namely CELLPU, CELLPD, and CELLΔ arrays.

Step 2: Start observing from the data of CELL 1 to CELL 18, moving from bottom to top. Search for certain characteristic values in CELLΔ data to determine whether open wire situations have occurred. Typically, for the starting position of a series of adjacent and consecutive open wires, if this starting position occurs at Cn pin ($n = 1$ to 17), then a voltage value greater than a positive threshold of +400 mV can be observed in the corresponding CELLΔ[n] data. This voltage value exceeding the threshold serves as a flag indicating the occurrence of an open wire at the Cn pin, signifying the likely starting point of a series of subsequent adjacent and consecutive open wire situations. Upon detecting this flag, proceed to Step 3.

Step 3: Starting from the flagged Cn pin position identified in Step 2, begin a loop to calculate the difference between $CELL\Delta[n]$ and $CELL\Delta[n+1]$ while incrementing the value of n, the maximum value of n should not exceed 17. This is the additional calculation step previously mentioned. The difference results exhibit certain characteristics during open wire detection: from the starting position of an open wire to subsequent adjacent and consecutive open wire positions, if the consecutive open wire situation is not interrupted, the difference between $CELL\Delta[n]$ and

¹ While reading this article, if readers combine observations of figures 5 and 6 to understand the algorithm, they will notice that in both figures, at adjacent and consecutive open wire locations, the difference between $CELL\Delta[n]$ and $CELL\Delta[n+1]$ is always either 0 or a positive number, without the occurrence of small negative values. This might seem strange. However, it is because the multiple circuit models in figures 5 and 6 have been simplified for easier calculation, resulting in more ideal data. Nevertheless, this simplification is acceptable for an approximate description of circuit behavior. In actual circuit testing for open wires, this difference could indeed be a small negative number, 0, or a positive number.

$CELL\Delta[n+1]$ will always be a very small negative value¹, 0, or a positive value. Like the positive threshold set in Step 2, a negative threshold of -400 mV is established for determining open wire situations in this step. During the loop calculation of the difference between $CELL\Delta[n]$ and $CELL\Delta[n+1]$, if the difference results are greater than -400 mV, it indicates an open wire situation at the C_n pin.

Step 4: Continuing the loop calculation described in Step 3. When the open wire condition ceases, notice that the difference between $CELL\Delta[n]$ and $CELL\Delta[n+1]$ is no longer greater than -400 mV, but usually becomes a very large negative value. Upon detecting that the result is smaller than -400 mV, it signifies the end position of a series of adjacent and consecutive open wires at the C_n pin. At this point, skip the previously detected open wire positions and repeat the calculations from Step 2 to Step 4 until all 18 cell channels and all open wire detections for C pins (excluding C0 pin and C18 pin) are completed.

Using examples illustrated in figures 5 and 6 for instances where multiple C pins experience simultaneous open wires, the above algorithm is rendered in a more concrete manner to help readers visually grasp the detection logic.

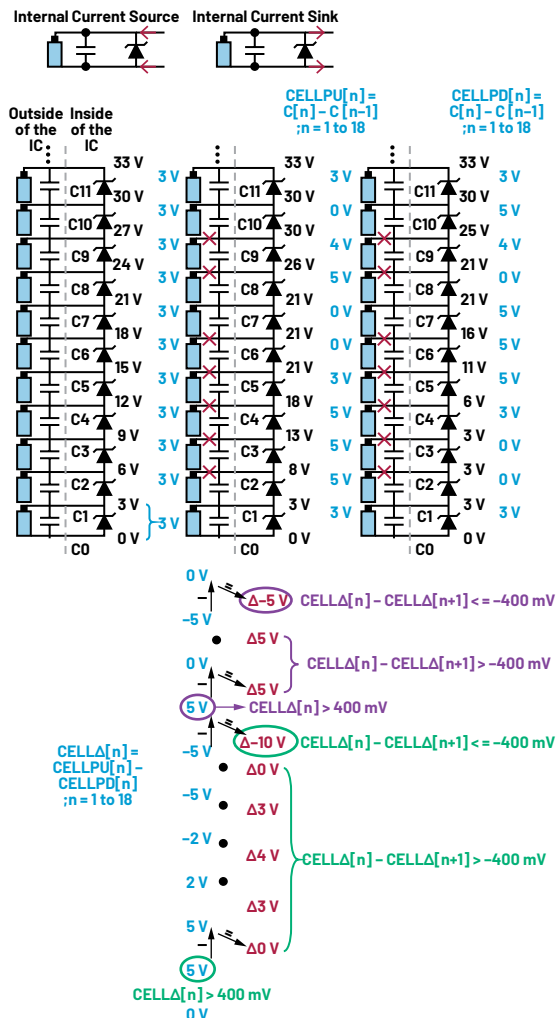


Figure 6. Detection algorithm diagram for multiple open wires occurring in the BMS

(Example 2).

Similarly, practical experiments and waveform tests for detecting multiple open wires on C pins were conducted using an LTC6813 evaluation board (the DC2350B) and 18650 batteries. This allowed for a clearer understanding of the LTC6813 circuit behavior during multiple open wire detection. The test results are presented in Figure 7 and Figure 8. In the circuit test of Figure 7, simultaneous open wires occurred at the C2 pin and C3 pin positions, while in the circuit test of Figure 8, simultaneous open wires occurred at the C2 pin, C3 pin, and C4 pin positions. Additionally, both figures provide the waveform test results of $CELL2/CELL3/CELL4$ under the corresponding open wire conditions during the execution of the ADOW command. From the test results, the observed changes in circuit behavior were consistent with the depiction in Figure 5 and Figure 6.

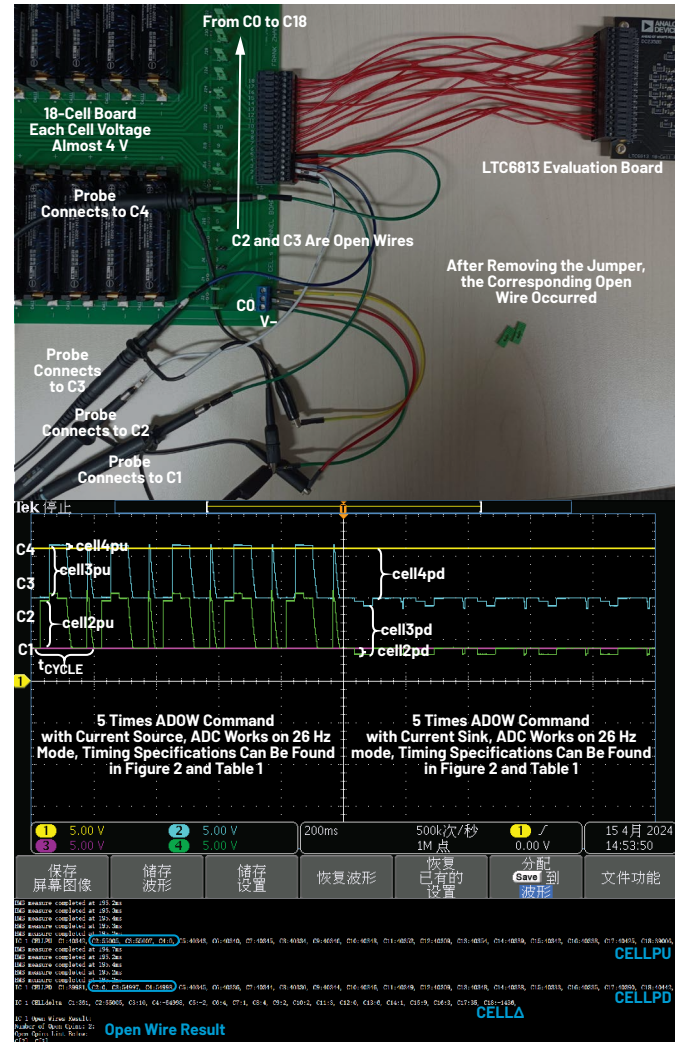


Figure 7. The experimental results of open wire detection and waveform testing (Experiment 3).

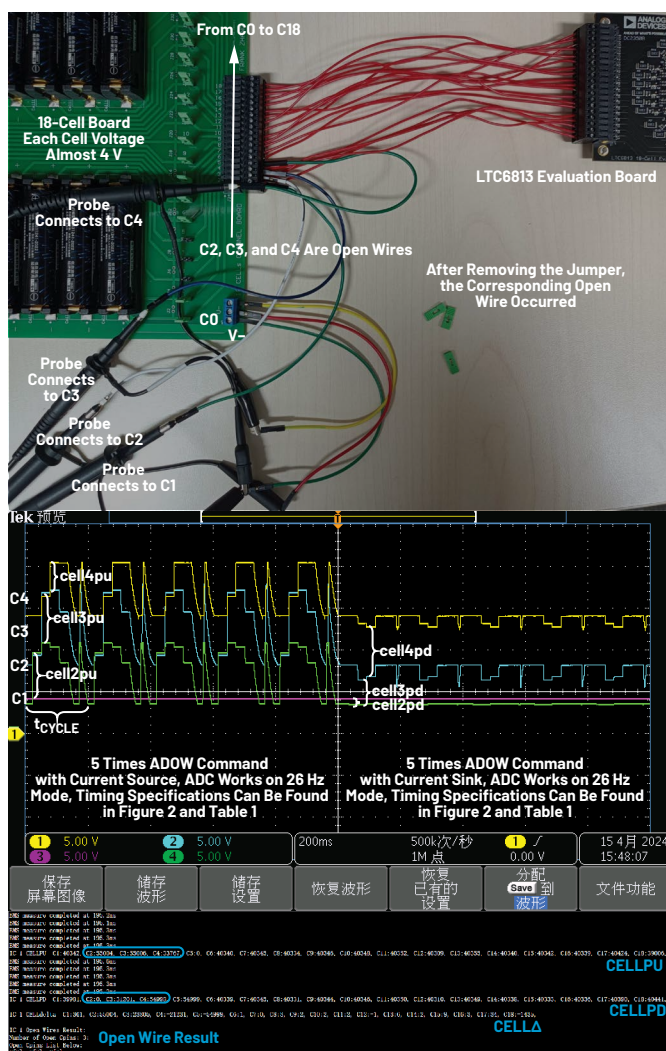


Figure 8. The experimental results of open wire detection and waveform testing (Experiment 4).

The algorithm is not only highly effective in detecting instances where multiple C pins experience simultaneous open wire faults, but it is also equally applicable to scenarios where only a single open wire occurs in the entire cell monitor. The reader can independently apply the above algorithm to detect open wires in situations where only one C pin is open (C0 pin and C18 pin excluded), and they will find that correct open wire detection results can still be obtained. The above algorithm not only inherits the advantages of the algorithm described in the data sheet but also overcomes the limitations of the data sheet algorithm when applied to situations where multiple C pins experience simultaneous open wires, making the detection of multiple open wire faults more accurate and efficient.

Situations When the C0 Pin or C18 Pin Experience an Open Wire

The algorithm pertains to detecting open wire situations in the LTC6813 from pins C1 to C17. Returning to the aforementioned algorithm, it is still very difficult to accurately detect all open wire faults in the circuit when the C0 pin or C18 pin experience an open wire situation. The following scenarios explore the complexity of the situations when these two pins experience open wires:

Scenario 1: As depicted in Figure 1, open wires at the C0 pin or the C18 pin can be assessed using the method mentioned in the data sheet, provided they do not occur simultaneously with adjacent C pins. Specifically, if there are no simultaneous open wires at the C0 pin and the C1 pin, or at the C18 pin and the C17 pin, the situation can be judged as follows: If $CELLPU[1] = 0$, then C0 is open. If $CELLPD[18] = 0$, then C18 is open.

Scenario 2: As shown in Figure 9 and Figure 10, when open wires occur simultaneously between the C0 pin or the C18 pin with their adjacent C pins, typically in such cases, the open wire situation for the C18 pin can still be identified using the condition if $CELLPD[18] = 0$, then C18 is open. However, when the C18 pin experiences simultaneous open wires with multiple adjacent C pins, it is recommended to set the open wire detection rule for the C18 pin as $CELLPD[18] < +400$ mV. Meanwhile, open wires adjacent to the C18 pin can still be identified using the previously mentioned algorithm.

As for the C0 pin, its open wire can be identified using the condition if $CELLPU[1] = 0$ then C0 is open. However, open wires adjacent to the C0 pin cannot be identified using the algorithm previously mentioned because they fail to meet the triggering conditions. For those open wires that cannot be identified by the algorithm, they can still be recognized through the open wire detection rule mentioned in the data sheet: if $CELL\Delta[n+1] < -400$ mV, then C(n) is open. However, it should be noted that for precise detection of the open wire of the C1 pin under Scenario 2, an additional open wire detection rule needs to be added as part of the algorithm previously mentioned: if C0 is open and $CELLPD[1] < 400$ mV, then C1 is open.

Scenario 3: As depicted in Figure 11, by shorting C0 to V- and C18 to V+ within the PCB, the advantage of this approach is the elimination of two wiring connections between the V- pin and the V+ pin and the battery pack, respectively. However, the disadvantage is that if the impedance of the wire connection between the C0 pin and the C18 pin is not sufficiently low, it may introduce additional errors to the voltage detection of CELL1 and CELL18. Because when C0 and V- and C18 and V+ share the same wiring connection, the V- pin and the V+ pin will carry the supply current during chip operation, which is typically significant. For instance, when the cell monitor's core is under MEASURE state and isoSPI is under READY state ($I_B = 1$ mA), according to the formulas listed in Table 2 and Table 3 from the LTC6813 data sheet, the current consumption can reach approximately 21 mA. If the wiring impedance is not sufficiently low, significant voltage drops may occur on the C0 and C18 wiring connection, thereby affecting the voltage detection accuracy of CELL1 and CELL18.

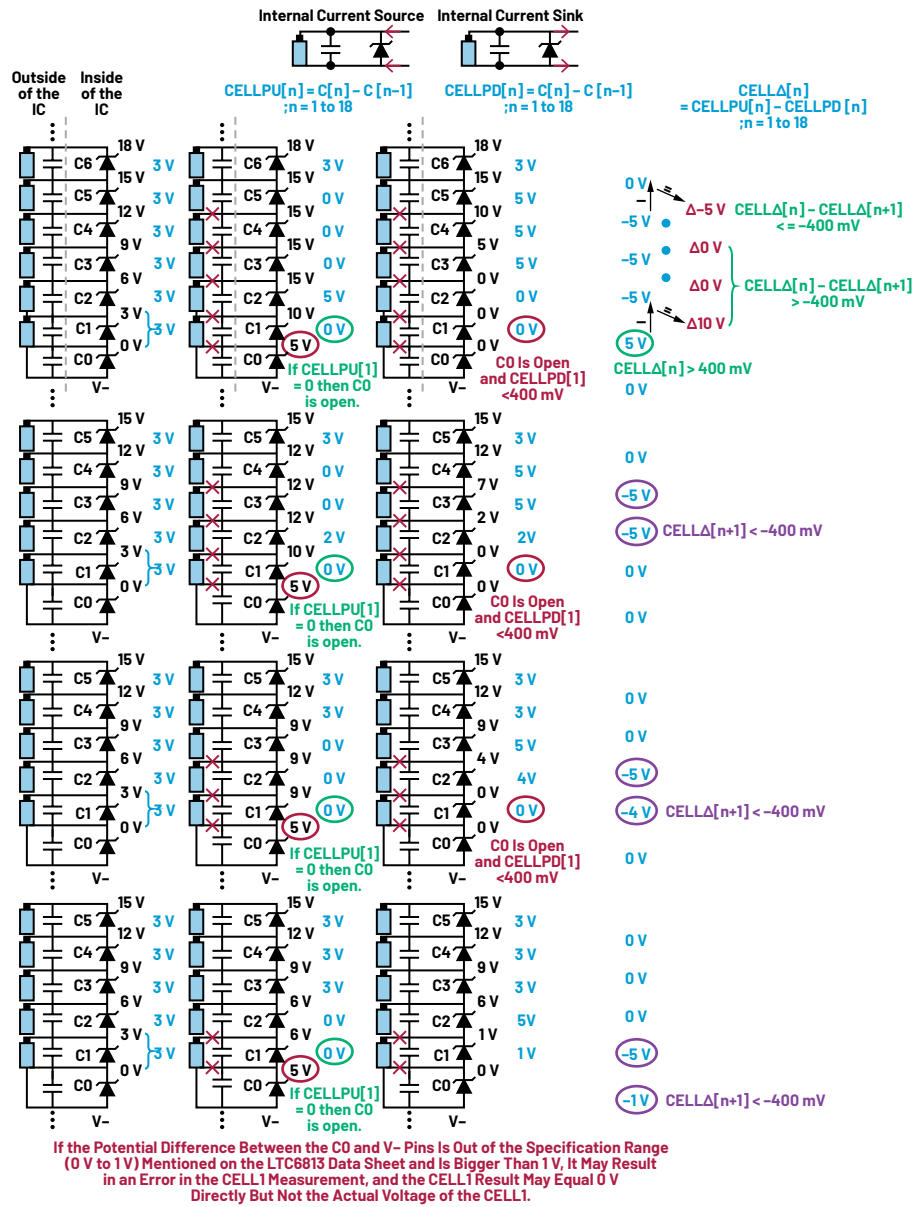


Figure 10. Detection algorithm diagram when the C0 pin experiences an open wire (based on Scenario 2).

Conclusion

Although this article mainly focuses on discussing the open wire detection methods and algorithms for the LTC6813, the open wire detection method used by this IC is also a common method employed in various models of ADI cell monitors and possesses generality. Therefore, the open wire detection algorithms described in this article can be easily transplanted to other ADI cell monitors that use similar open wire detection methods with just minor modifications.

```
N_CHANNELS=18;
// number of ltc6813 cell channels
CELLPU[N_CHANNELS];
CELLPD[N_CHANNELS];
CELLΔ[N_CHANNELS];
// three array to store the cell voltage data
OPENC Pins[N_CHANNELS+1];
// array to store the open wires
WAKEUPIC();
// wake up the ltc6813
CLRCELL();
// clears cell voltage register
for(I=0; I<5; I++)
{
    WAKEUPIC();
    ADOW(PULL_UP_CURRENT);
    // execute the adown command 5 times with current source
    POLLADC();
    //poll adc converter status
}
WAKEUPIC();
CELLPU[N_CHANNELS]=RDCV();
//store adc conversion results into array cellpu[n]
for(I=0; I<5; I++)
{
    WAKEUPIC();
    ADOW(PULL_DOWN_CURRENT);
    // execute the adown command 5 times with current sink
    POLLADC();
}
WAKEUPIC();
CELLPD[N_CHANNELS]=RDCV();
//store adc conversion results into array cellpd[n]
for(I=0; I<N_CHANNELS; I++)
{
    CELLΔ[I]=CELLPU[I]-CELLPD[I];
    //calculate the cellΔ result
}
N=0;
J=0;
for(CELL=0; CELL<N_CHANNELS; CELL++)
// start observing from the data of cell 1 to cell 18
// moving from bottom to top
{
    if((CELL<=J)&&(CELL!=0)&&(CELL!=(N_CHANNELS-1)))
    {
        continue;
        // skip the previously detected open wire positions and repeat the calculations
    }
    else if((PULLPU[CELL]==0)&&(CELL==0))
    {
        OPENC Pins[N]=CELL;
        // if cellpu[0] =0V then C0 is open.
        N++;
        if(PULLPD[CELL]<4000)
        {
```

```
        else if((CELLΔ[CELL]<-4000))
        {
            OPENC Pins[N]=CELL;
            // if cellΔ[n] < -400mV, then C(n) is open
            N++;
        }
    }
    else if(CELLΔ[CELL]>4000)
    {
        //cellΔ[n]>400mV signifying the likely starting point
        //of a series of subsequent "adjacent and consecutive" open wire situations
        for(J=CELL;J<N_CHANNELS-1;J++)
        {
            if((CELLΔ[J]-CELLΔ[J+1])>-4000)
            {
                OPENC Pins[N]=J+1;
                //continuing the loop calculation of cellΔ[n] - cellΔ[n+1],
                // large then -400mV indicates an open wire situation at the C(n+1) pin
                N++;
            }
            else
                break;
        }
    }
    if((PULLPD[CELL]<4000)&&(CELL==(N_CHANNELS-1)))
    {
        OPENC Pins[N]=CELL+1;
        //pullpd[17]<400mV then C18 is open
        N++;
        for(K=CELL-1;K>=0;K--)
        {
            if((PULLPU[K]<4000)&&(PULLPD[K]<4000))
            {
                OPENC Pins[N]=K+1;
                // if C18 is open,
                // and If cellpu[n] < +400mV && cellpd[n] < +400mV then C(n+1) is open
                N++;
            }
            else
                break;
        }
    }
}
REMOVE_REPETITIVE(OPENC Pins[N]);
//removing repetitive elements of open wire array
SORT_ARRAY(OPENC Pins[N]);
// sorting open wire array
```

Figure 13. Pseudocode example for open wire detection in the LTC6813 devices.

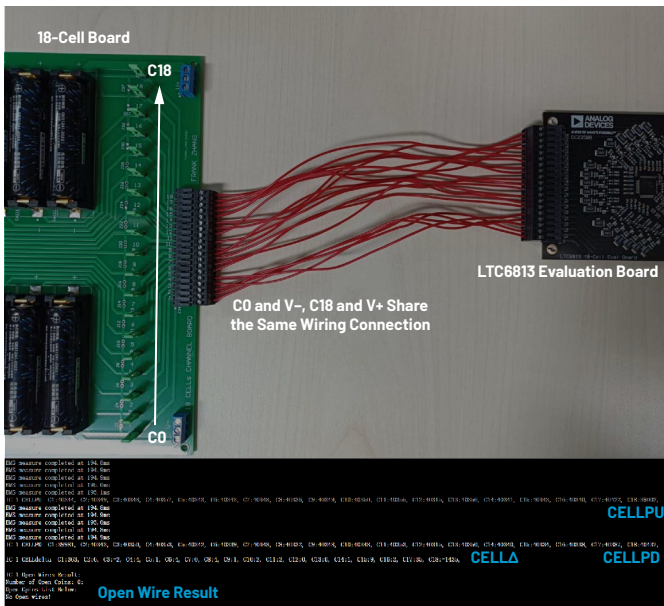


Figure 14. The open wire experiment and results (Experiment 1).

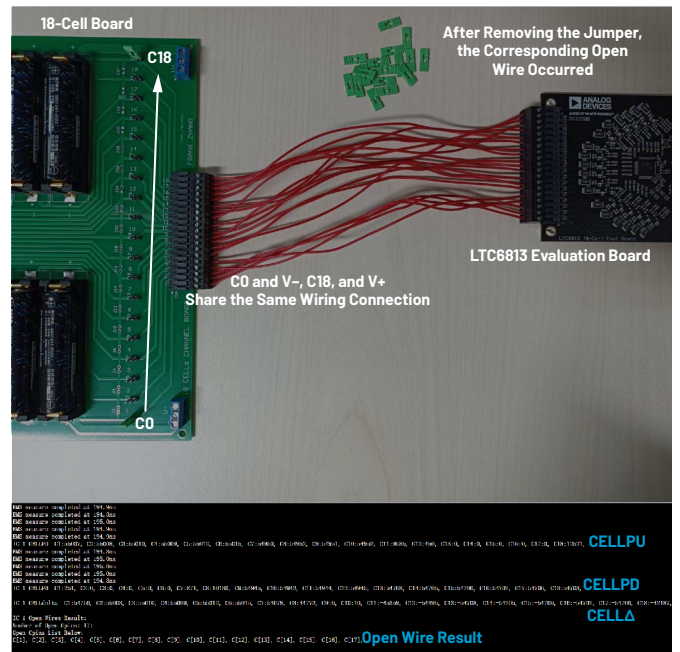


Figure 16. The open wire experiment and results (Experiment 3).

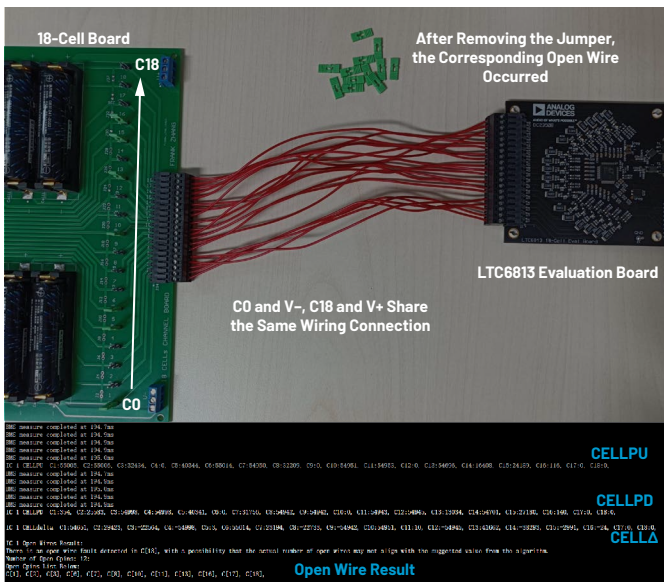


Figure 15. The open wire experiment and results (Experiment 2).

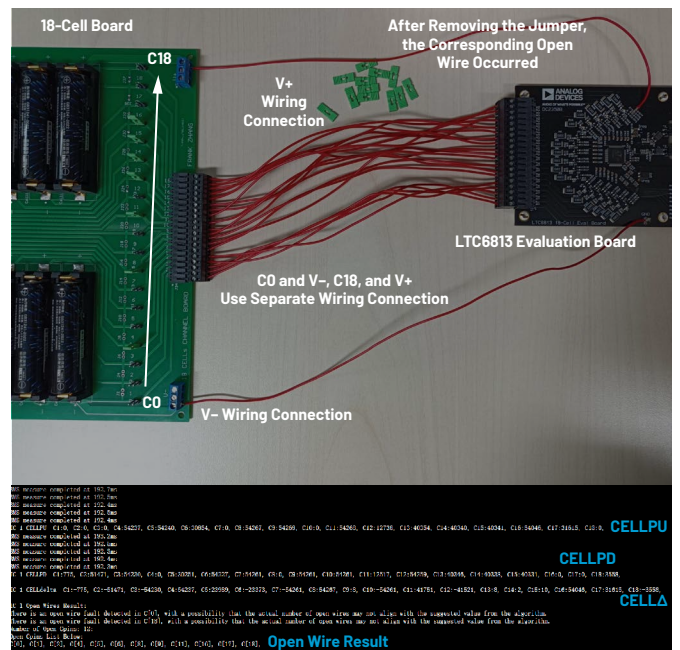


Figure 17. The open wire experiment and results (Experiment 4).

About the Author

Frank Zhang is an application engineer working in Central Applications China at Analog Devices. His areas of expertise are battery management systems (BMS), precision signal chains, and embedded software development. He received his M.S.E.E. degree from Fuzhou University in 2022 and joined ADI in the same year.

Engage with the ADI technology experts in our online support community. Ask your tough design questions, browse FAQs, or join a conversation.



Visit ez.analog.com