MCP79410
Hours Alarm Issue

Amond Lin

2024 SEP

Background

- **Application: Battery Life Calculation**
  - Using MCP79410 hours alarm output function to turn on the VDD power circuit of the external MCU, and only the RTC chip is powered continuity.
  - Once the MCU is powered up, the MCU will read the ALM0HOUR and then write back with decreasing by 1. And the **23-hour alarm counter** of MCU would be increased 1.
  - So, the MCU would be powered up every 23 hours triggered by MCP79410 RTC alarm output function.
  - Once **23-hour alarm counter** reaches max value (~5 years), it means battery is end of live.

Initialization

- RTCSEC=0x00
- RTCMIN=0x00
- RTCHOUR=0x13
- RTCWKDAY=0x0B
- RTCDATE=0x22
- RTCMTH=0x06
- RTCYEAR=0x22
- CONTROL=0x10
- RTCSEC=0x80

- ALM0SEC=0x00
- ALM0MIN=0x00
- ALM0HOUR=0x00
- ALM0WKDAY=0x23

Hour match

Right after production & Normal operation

- 1st step of their production SOP was to configure RTC based on present date & time and to configure 1st Alarm time with 0:00.
- Once the 1st alarm occurred, ALM0HOUR would be read and written back with decreasing 1.
- And then so on, the procedure would repeat every 23 hours.

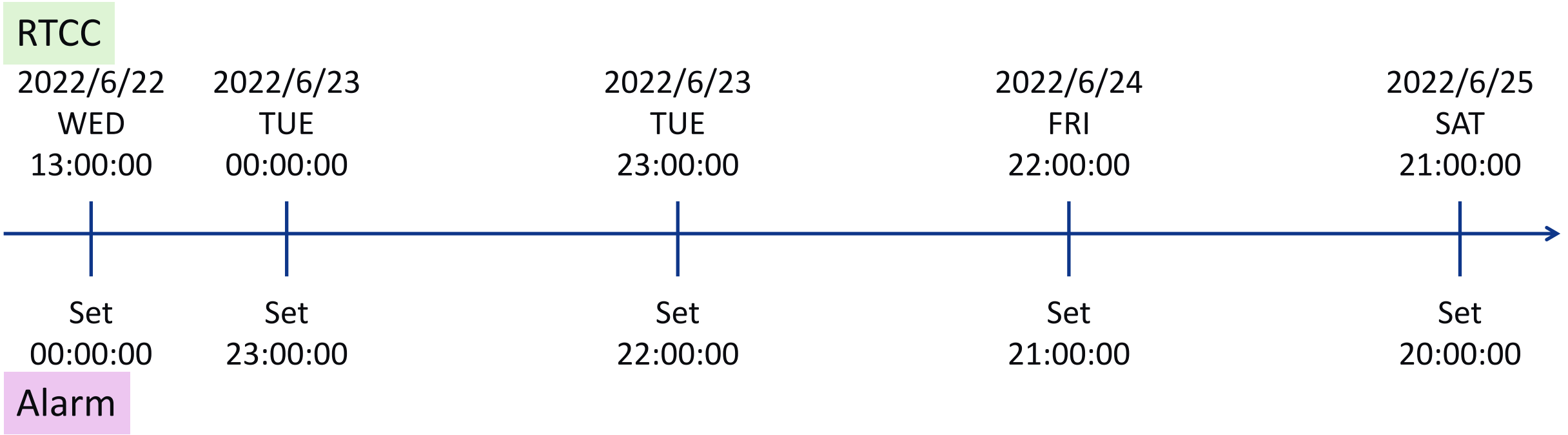
Failure Description

- **For example with above initial settings,**
  - MCP79410 RTC alarm output function would occur when RTCHOUR counted to 0x21 (ALM0HOUR = 0x21).
  - At the moment, the MCU would be powered up and read the ALM0HOUR (= 0x21) and then write back with 0x20.
  - It was expected to get alarm again after 23 hours, but the failure chips would alarm the system just after one hour (RTCHOUR=0x22) and then alarm the system again after 23 hours RTCHOUR=0x20).
  - There should be only one-time alarm in 23 hours, but there were 2 times recorded.
- **Additional information:**
  - The failure rate was about 20% so far.
  - One of failure chip:
    - Microchip MCP79410-I/ST
    - Track code.7941
    - Date: I307 (Y2023, 07W)
  - The host MCU: dsPIC33CK128MP206

Failure Analysis (Based on data packet from client)

- **Design operation**
  - Initialized date/time with present date/time (2022, JUN, 22, Wednesday, 13:00:00)
  - **Configure 1st Alarm time at 0:00**.
  - When ST(RTCSEC bit <7>)=1, RTCC would start working.
  - When hour-match at 00:00:00, ALM0IF=1, MCU would config ALM0HOUR=0x23 and clear ALM0IF.
  - When next time with ALM0IF=1, MCU would be powered up to read and write back ALM0HOUR (with decreasing 1) and then clear ALM0IF. And so on, the procedure would be expected with repeating every 23 hours.

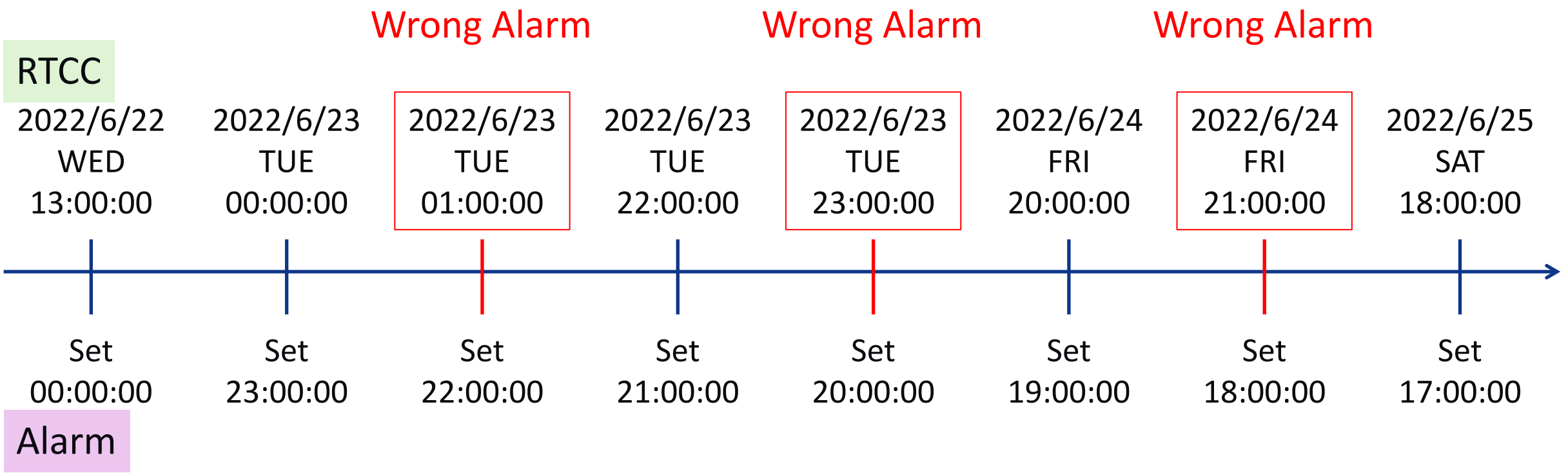
Time-line：Design operation



Failure Analysis (Based on data packet from client)

- **The problem:**
  - Initialized date/time with present date/time (2022, JUN, 22, Wednesday, 13:00:00)
  - **Configure 1st Alarm time at 0:00**.
  - When ST(RTCSEC bit <7>)=1，RTCC would start working.
  - When hour-match at 00:00:00，ALM0IF=1.
  - MCU would config ALM0HOUR=0x23 and clear ALM0IF.
  - It was expected to get next alarm (let’s call 2nd alarm) after 23 hours, but 2nd aleam would occur again after only 1 hour (RTCHOUR=0x01). MCU didn’t know that, so MCU read and wrote back ALM0HOUR (0x23-1=0x22) again and clear ALM0IF.
  - 3rd alarm would occur properly after 23 hours.
  - 4th alarm would occur after only hour again.
  - 5th alarm would occur properly after 23 hours.
  - 6th alarm would occur after only hour again.
  - And so so…until MCP79410 was power-down.

Time-line：The problem



Sept-20, 2024

Reported by Weikeng Amond Lin

All tests were based on Weikeng’s test environment and conditions.

Test process

- **Initializations:**
  - RTCSEC=0x00
  - RTCMIN=0x00
  - RTCHOUR=0x05
  - RTCWKDAY=0x03
  - RTCDATE=0x18
  - RTCMTH=0x09
  - RTCYEAR=0x24
  - CONTROL=0x14
  - ALM0SEC=0x00
  - ALM0MIN=0x00
  - ALM0HOUR=0x10
  - ALM0WKDAY=0x20
  - RTCSEC=0x80

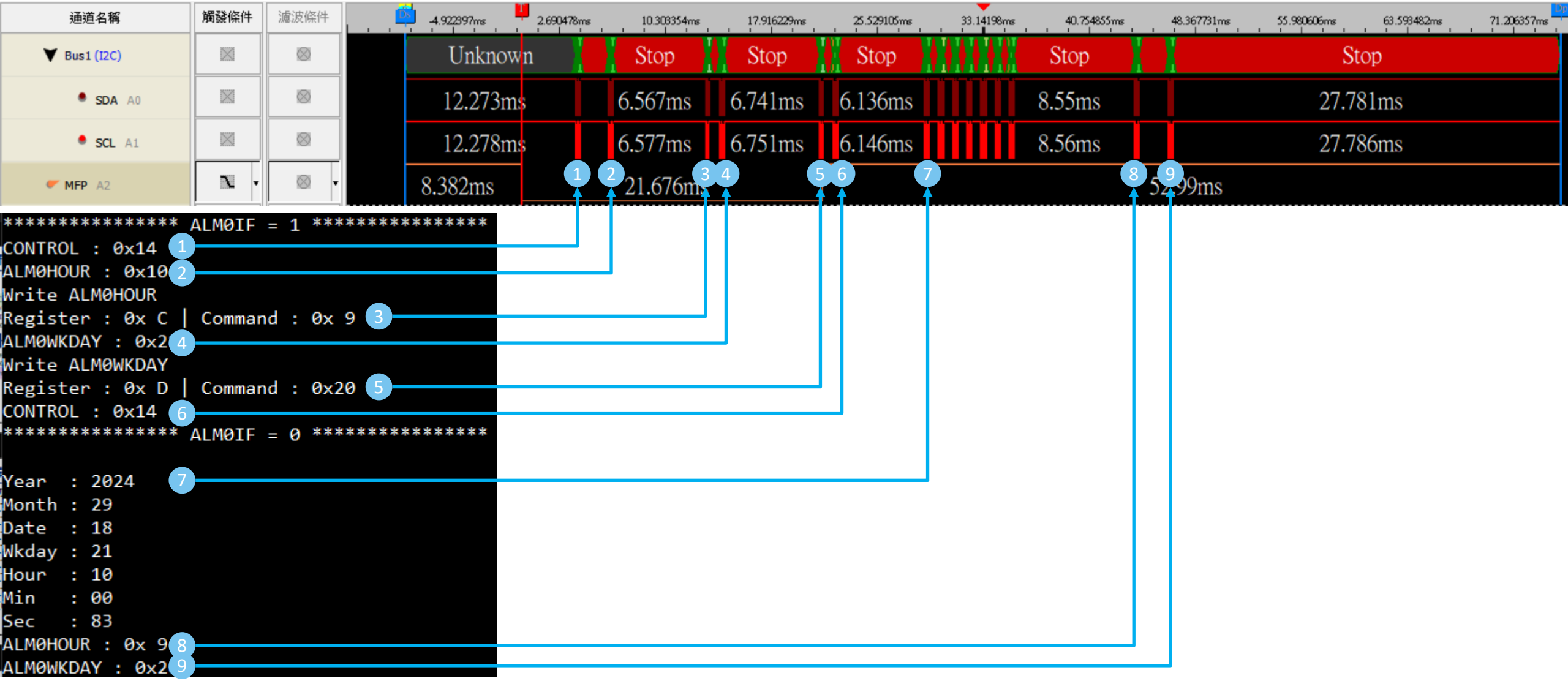
Test process

- **dsPIC33CK256MP506 would check MCP79410 one time once MFP pin triggered MCU.**
- **To speed up issue reproduction**
  - 0x07H: CRSTRIM = 1
  - 0x08H: OSCTRIM=0xFF
  - ~10mins in real world would be about 23 hours of RTCC.

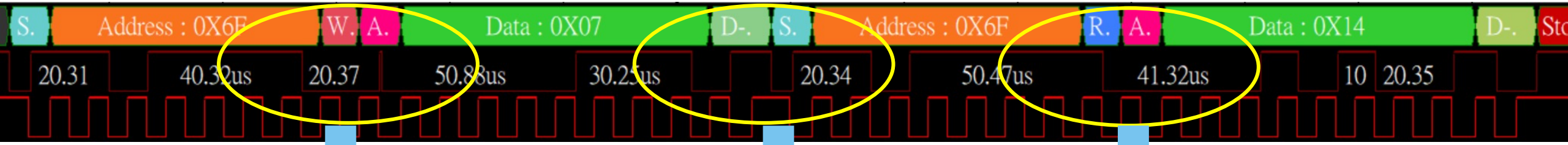
Data packet (Duplicated on EVM, not from client)

封包	名稱	起始點	Start	Address	Write	A-ACK	Data	D-ACK			
1	Bus1(I2C)	3.89079ms	Start	6F	Write	A-ACK	07	D-ACK			
2	Bus1(I2C)	4.09025ms	Start	6F	Read	A-ACK	14	D-NACK	Stop	Master NACK	
3	Bus1(I2C)	6.28412ms	Start	6F	Write	A-ACK	0C	D-ACK			
4	Bus1(I2C)	6.48357ms	Start	6F	Read	A-ACK	10	D-NACK	Stop	Master NACK	
5	Bus1(I2C)	13.25091ms	Start	6F	Write	A-ACK	0C	D-ACK	09	D-ACK	Stop
6	Bus1(I2C)	14.26477ms	Start	6F	Write	A-ACK	0D	D-ACK			
7	Bus1(I2C)	14.46427ms	Start	6F	Read	A-ACK	28	D-NACK	Stop	Master NACK	
8	Bus1(I2C)	21.406ms	Start	6F	Write	A-ACK	0D	D-ACK	20	D-ACK	Stop
9	Bus1(I2C)	22.41982ms	Start	6F	Write	A-ACK	07	D-ACK			
10	Bus1(I2C)	22.61933ms	Start	6F	Read	A-ACK	14	D-NACK	Stop	Master NACK	
11	Bus1(I2C)	28.95579ms	Start	6F	Write	A-ACK	00	D-ACK			
12	Bus1(I2C)	29.15526ms	Start	6F	Read	A-ACK	83	D-NACK	Stop	Master NACK	
13	Bus1(I2C)	29.97238ms	Start	6F	Write	A-ACK	01	D-ACK			
14	Bus1(I2C)	30.17176ms	Start	6F	Read	A-ACK	00	D-NACK	Stop	Master NACK	
15	Bus1(I2C)	30.98901ms	Start	6F	Write	A-ACK	02	D-ACK			
16	Bus1(I2C)	31.18843ms	Start	6F	Read	A-ACK	10	D-NACK	Stop	Master NACK	
17	Bus1(I2C)	32.0057ms	Start	6F	Write	A-ACK	03	D-ACK			
18	Bus1(I2C)	32.20512ms	Start	6F	Read	A-ACK	23	D-NACK	Stop	Master NACK	
19	Bus1(I2C)	33.02315ms	Start	6F	Write	A-ACK	04	D-ACK			
20	Bus1(I2C)	33.22271ms	Start	6F	Read	A-ACK	18	D-NACK	Stop	Master NACK	
21	Bus1(I2C)	34.03961ms	Start	6F	Write	A-ACK	05	D-ACK			
22	Bus1(I2C)	34.23898ms	Start	6F	Read	A-ACK	29	D-NACK	Stop	Master NACK	
23	Bus1(I2C)	35.05597ms	Start	6F	Write	A-ACK	06	D-ACK			
24	Bus1(I2C)	35.25542ms	Start	6F	Read	A-ACK	24	D-NACK	Stop	Master NACK	
25	Bus1(I2C)	44.00665ms	Start	6F	Write	A-ACK	0C	D-ACK			
26	Bus1(I2C)	44.20609ms	Start	6F	Read	A-ACK	09	D-NACK	Stop	Master NACK	
27	Bus1(I2C)	46.48592ms	Start	6F	Write	A-ACK	0D	D-ACK			
28	Bus1(I2C)	46.68535ms	Start	6F	Read	A-ACK	20	D-NACK	Stop	Master NACK	

Read/Write with UART print info



Data packet timing



Sept-22, 2024

Reported by Weikeng Amond Lin

All tests were based on Weikeng’s test environment and conditions.

Typo – Update p.12

- P.12
  - 0x07H: CRSTRIM = 1
  - 0x08H: OSCTRIM=0xFF

Additional Feedback from client

- MCU would config CRSTRIM = 0 by byte-writing during initialization.
- OSCTRIM was never configured by MCU during/after initialization.

07h	CONTROL	OUT	SQWEN	ALM1EN	ALM0EN	EXTOSC	CRSTRIM	SQWFS1	SQWFS0
08h	OSCTRIM	SIGN	TRIMVAL6	TRIMVAL5	TRIMVAL4	TRIMVAL3	TRIMVAL2	TRIMVAL1	TRIMVAL0
09h	EEUNLOCK	Protected EEPROM Unlock Register (not a physical register)							

Additional Feedback from client

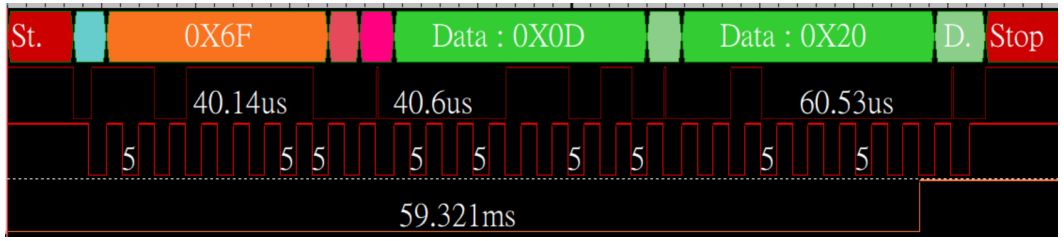
- I2C speed: 100K with 3.3V Vdd
- I2C AC CHARACTERISTICS

			1300	—	—	ns	2.5V ≤ Vcc ≤ 5.5V
4	TR	SDA and SCL Rise Time (Note 1)	—	—	1000	ns	1.8V ≤ Vcc < 2.5V
			—	—	300	ns	2.5V ≤ Vcc ≤ 5.5V

  - Measured Tr = 450ns

Question from Weikeng to client

- Q: Once ALM0IF was cleared, MFP would be pull-high immediately. Was the last frame sent completely before MCU was off?



- Ans:
  - Client will compare the last frame timing with MFP pin.

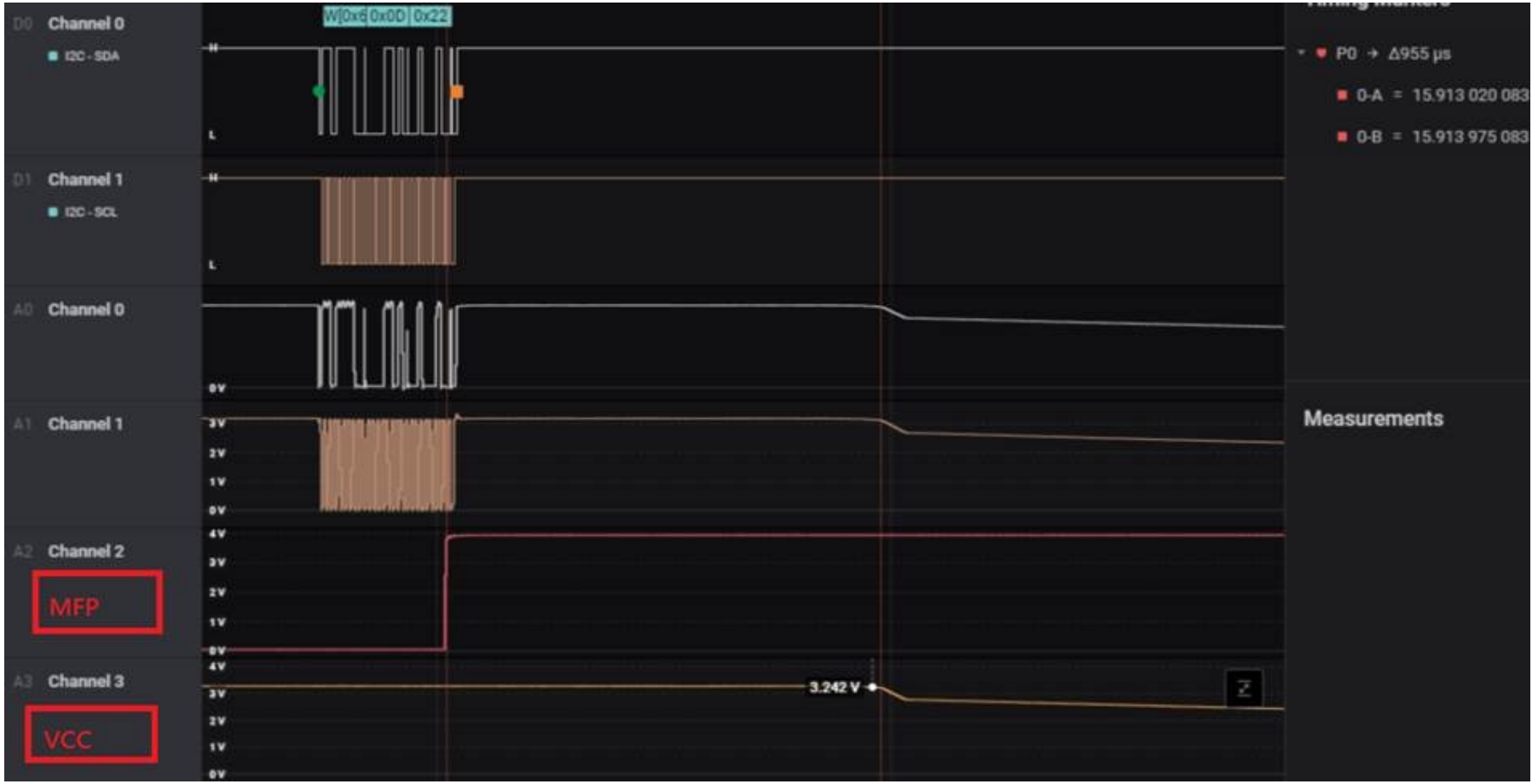
Sept-23, 2024

Reported by Weikeng Amond Lin

All tests were based on Weikeng’s test environment and conditions.

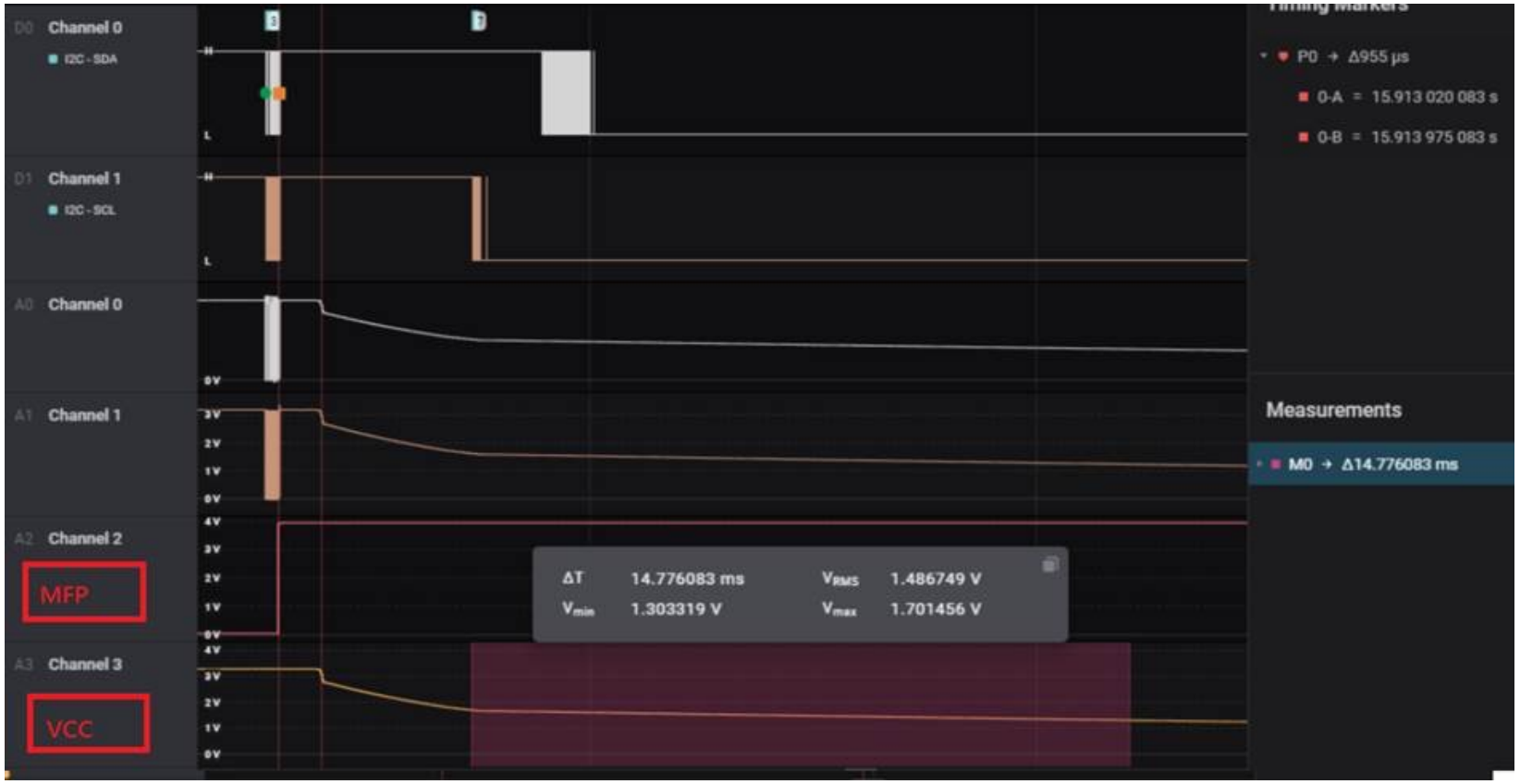
Timing sequence from client (1/3)

- VCC would start falling down after 955ns with MFP pull-high.



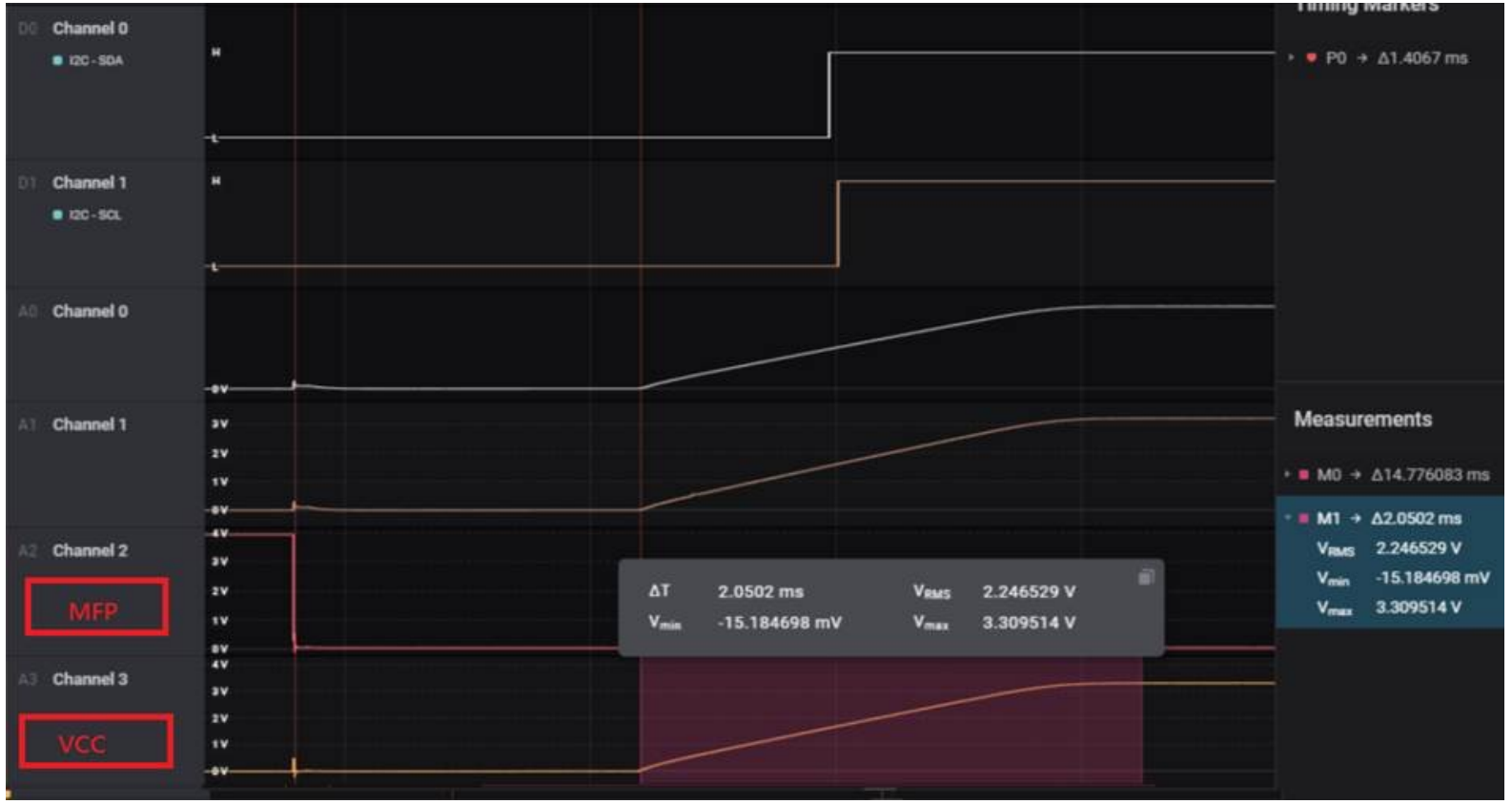
Timing sequence from client (2/3)

- When VCC was powered down (RTC got to sleep), VCC down from 1.7V to 1.3V would take about 14ms to compare with T15 on data sheet.



Timing sequence from client (3/3)

- When RTCC waked up, MFP was low. Right after 14ms, VCC started going up from 0 to 3.3V to compare with T16 on datasheet.



Test process

- Initializations:

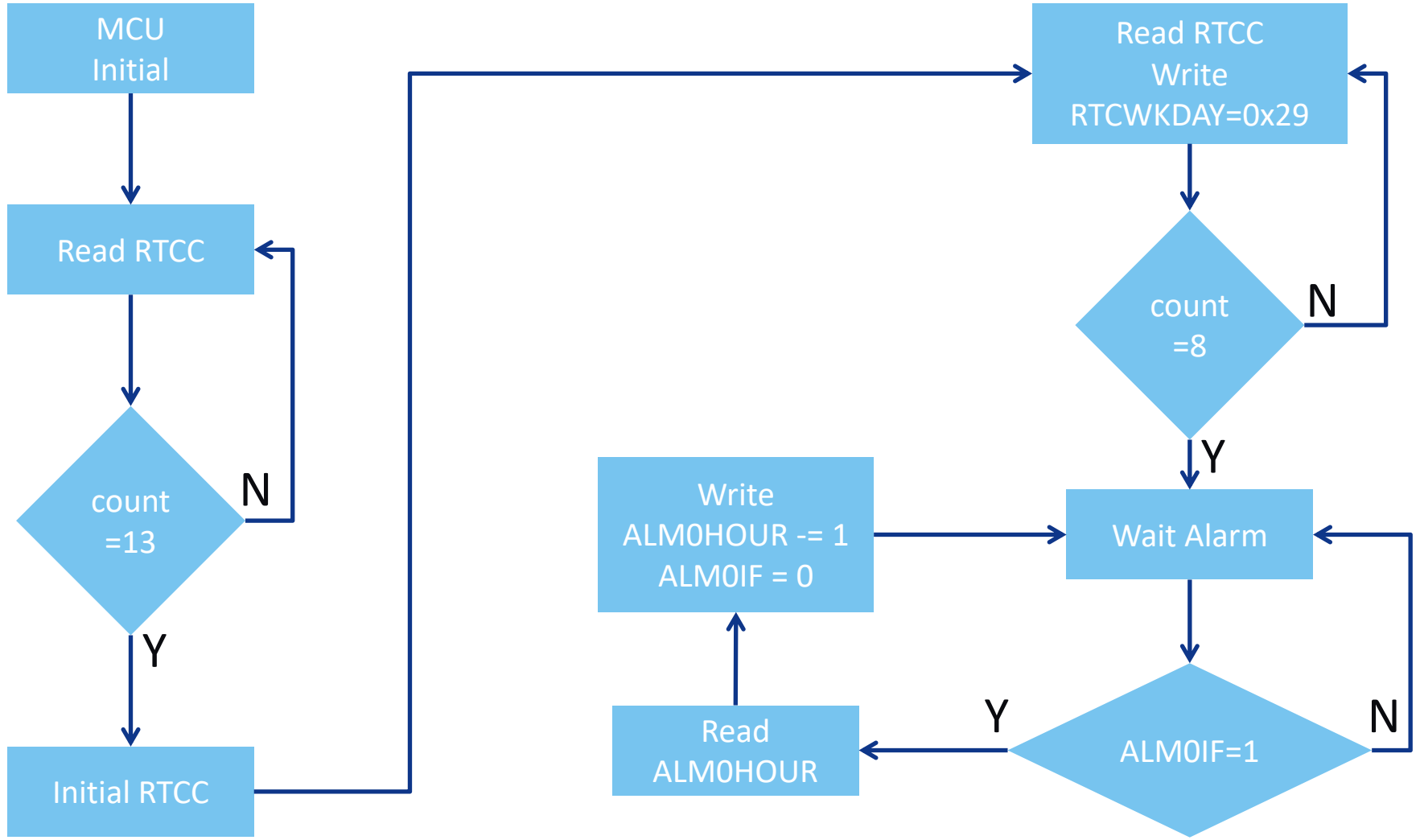
```
cmd MCP79410_initCmd[] = {
    /* 0x00 */  {RTCSEC,     0x30,                          "RTCSEC"},
    /* 0x01 */  {RTCMIN,     0x06,                          "RTCMIN"},
    /* 0x02 */  {RTCHOUR,    0x16,                          "RTCHOUR"},
    /* 0x03 */  {RTCWKDAY,   (PWRFAIL|VBATEN|WKDAY),       "RTCWKDAY"},
    /* 0x04 */  {RTCDATE,    0x18,                          "RTCDATE"},
    /* 0x05 */  {RTCMTH,     0x09,                          "RTCMTH"},
    /* 0x06 */  {RTCYEAR,    0x23,                          "RTCYEAR"},
    /* 0x07 */  {CONTROL,    (ALM1EN|ALM0EN|EXTOSC|CRSTRIM), "CONTROL"}, // ALM0EN=1, CRSTRIM=1
    /* 0x08 */  {OSCTRIM,    0xFF,                          "OSCTRIM"},
    /* 0x09 */  {EEUNLOCK,   0x00,                          "EEUNLOCK"},
    /* 0x0A */  {ALM0SEC,    0x00,                          "ALM0SEC"},
    /* 0x0B */  {ALM0MIN,    0x00,                          "ALM0MIN"},
    /* 0x0C */  {ALM0HOUR,   0x00,                          "ALM0HOUR"},
    /* 0x0D */  {ALM0WKDAY,  (ALMMSK|ALM0IF|WKDAY),         "ALM0WKDAY"},
    /* 0x0E */  {ALM0DATE,   0x00,                          "ALM0DATE"},
    /* 0x0F */  {ALM0MTH,    0x00,                          "ALM0MTH"},
};
```

```
#define ST          1 << 7
#define WKDAY       2 << 0
#define PWRFAIL     0 << 4
#define VBATEN      1 << 3
#define CRSTRIM     1 << 2
#define EXTOSC      0 << 3
#define ALM0EN      1 << 4
#define ALM1EN      0 << 5
#define ALMMSK      2 << 4
#define ALM0IF      0 << 3
```

Test process 2

- **Calibration process**
  - repeat read those register 14 times
    - ALM0WKDAY, RTCWKDAY, RTCSEC, RTCMIN, RTCHOUR, RTCDATE, RTCMTH, RTCYEAR, CONTROL, ALM0HOUR, ALM0MIN, ALM0SEC
  - Initial the following register in order
    - RTCSEC=0x30, ALM0WKDAY=0x22, ALM0SEC=0x00, ALM0MIN=0x00, ALM0HOUR=0x00, CONTROL=0x14, OSCTRIM=0xFF, RTCYEAR=0x23, RTCMTH=0x09, RTCDATE=0x18, RTCHOUR=0x16, RTCMIN=0x06, RTCWKDAY=0x0A, RTCSEC=0xB0
- **Power on sequence**
  - We programed the MCU and then disconnected the USB cable. Once VDD of the MCU and RTCC dropped to zero, the USB cable was connected again to check the initial flow.

Calibration Flowchart



Need to check

MCP79410 Initinal

RTCSEC 0x 0 = 0xB0
RTCMIN 0x 1 = 0x 6
RTCHOUR 0x 2 = 0x16
RTCWKDAY 0x 3 = 0x A
RTCDATE 0x 4 = 0x18
RTCMTH 0x 5 = 0x 9
RTCYEAR 0x 6 = 0x23
CONTROL 0x 7 = 0x14
OSCTRIM 0x 8 = 0xFF
EEUNLOCK 0x 9 = 0x 0
ALM0SEC 0x A = 0x 0
ALM0MIN 0x B = 0x 0
ALM0HOUR 0x C = 0x 0
ALM0WKDAY 0x D = 0x22

After RTCC initial and set ST bit(RTCSEC<Bit 7>)
RTCWKDAY = 0x0A

Index : 0

RTCSEC 0x 0 = 0xB0
RTCMIN 0x 1 = 0x 6
RTCHOUR 0x 2 = 0x16
RTCWKDAY 0x 3 = 0x29
RTCDATE 0x 4 = 0x18
RTCMTH 0x 5 = 0x 9
RTCYEAR 0x 6 = 0x23
CONTROL 0x 7 = 0x14
OSCTRIM 0x 8 = 0xFF
EEUNLOCK 0x 9 = 0x 0
ALM0SEC 0x A = 0x 0
ALM0MIN 0x B = 0x 0
ALM0HOUR 0x C = 0x 0
ALM0WKDAY 0x D = 0x2A

Upon reading the RTCC for the second time, we wrote RTCWKDAY=0x29 and found that ALM0IF had been set.

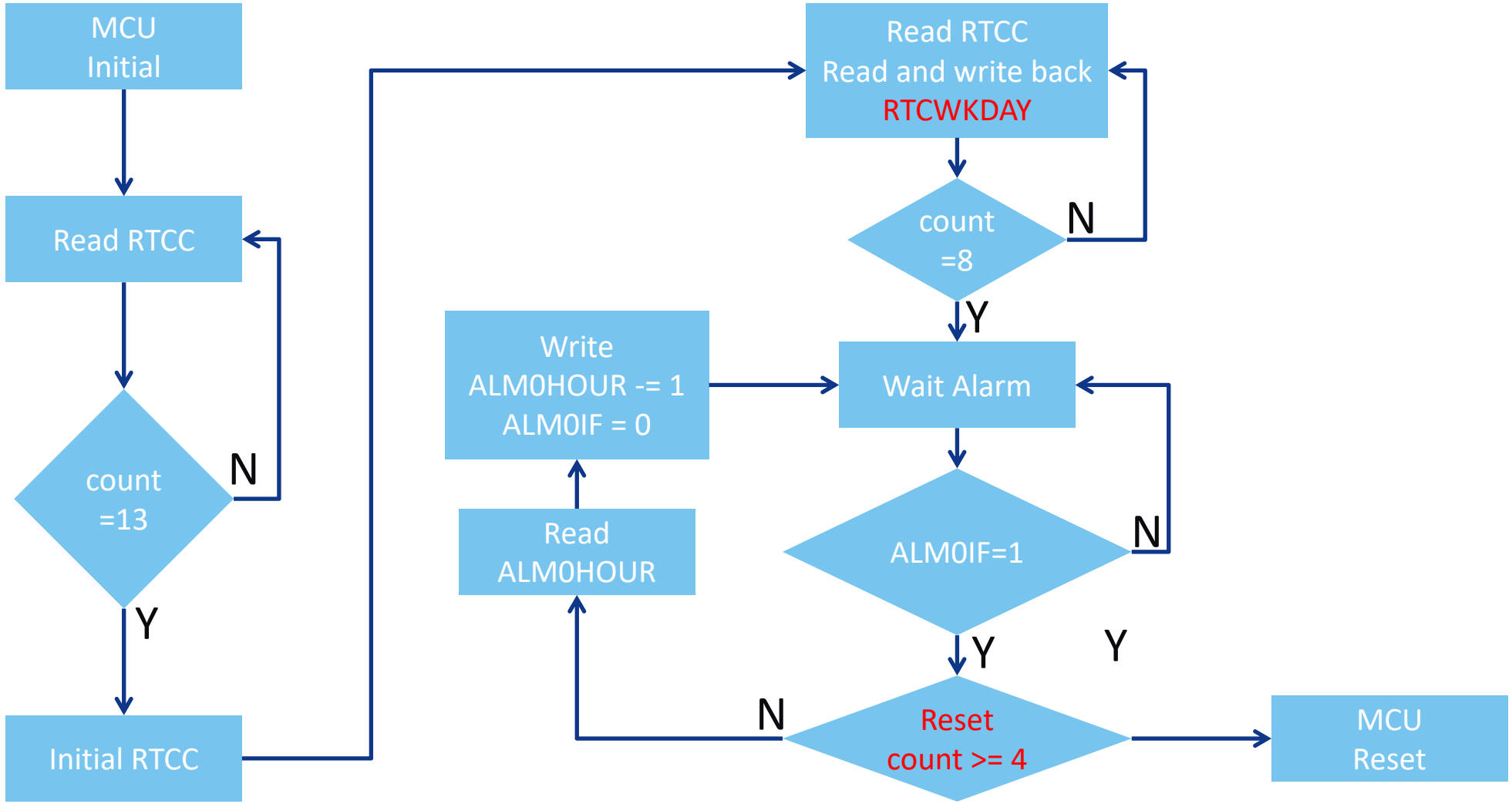
Only read for check PWRFAIL and write back, not modify value.

Sept-24, 2024

Reported by Weikeng Amond Lin

All tests were based on Weikeng’s test environment and conditions.

Calibration Flowchart：Modified to match actual process



Teraterm Log

```
RTCSEC 0x 0 = 0x84
RTCMIN 0x 1 = 0x 0
RTCHOUR 0x 2 = 0x 0
RTCWKDAY 0x 3 = 0x2B
RTCDATE 0x 4 = 0x19
RTCMTH 0x 5 = 0x 9
RTCYEAR 0x 6 = 0x23
CONTROL 0x 7 = 0x14
OSCTRIM 0x 8 = 0xFF
EEUNLOCK 0x 9 = 0x 0
ALM0SEC 0x A = 0x 0
ALM0MIN 0x B = 0x 0
ALM0HOUR 0x C = 0x 0
ALM0WKDAY 0x D = 0x2A
********** ALM0IF = 1 **********
ALM0HOUR : 0x 0
RTCHOUR : 0x 0
Count Rest : 1
Write ALM0HOUR
Register : 0x C | Command : 0x23
Write ALM0WKDAY
Register : 0x D | Command : 0x20
********** ALM0IF = 0 **********
```

```
RTCSEC 0x 0 = 0x84
RTCMIN 0x 1 = 0x 0
RTCHOUR 0x 2 = 0x23
RTCWKDAY 0x 3 = 0x2B
RTCDATE 0x 4 = 0x19
RTCMTH 0x 5 = 0x 9
RTCYEAR 0x 6 = 0x23
CONTROL 0x 7 = 0x14
OSCTRIM 0x 8 = 0xFF
EEUNLOCK 0x 9 = 0x 0
ALM0SEC 0x A = 0x 0
ALM0MIN 0x B = 0x 0
ALM0HOUR 0x C = 0x23
ALM0WKDAY 0x D = 0x28
********** ALM0IF = 1 **********
ALM0HOUR : 0x23
RTCHOUR : 0x23
Count Rest : 2
Write ALM0HOUR
Register : 0x C | Command : 0x22
Write ALM0WKDAY
Register : 0x D | Command : 0x20
********** ALM0IF = 0 **********
```

```
RTCSEC 0x 0 = 0x84
RTCMIN 0x 1 = 0x 0
RTCHOUR 0x 2 = 0x22
RTCWKDAY 0x 3 = 0x2C
RTCDATE 0x 4 = 0x20
RTCMTH 0x 5 = 0x 9
RTCYEAR 0x 6 = 0x23
CONTROL 0x 7 = 0x14
OSCTRIM 0x 8 = 0xFF
EEUNLOCK 0x 9 = 0x 0
ALM0SEC 0x A = 0x 0
ALM0MIN 0x B = 0x 0
ALM0HOUR 0x C = 0x22
ALM0WKDAY 0x D = 0x28
********** ALM0IF = 1 **********
ALM0HOUR : 0x22
RTCHOUR : 0x22
Count Rest : 3
Write ALM0HOUR
Register : 0x C | Command : 0x21
Write ALM0WKDAY
Register : 0x D | Command : 0x20
********** ALM0IF = 0 **********
```

Question from Weikeng to client

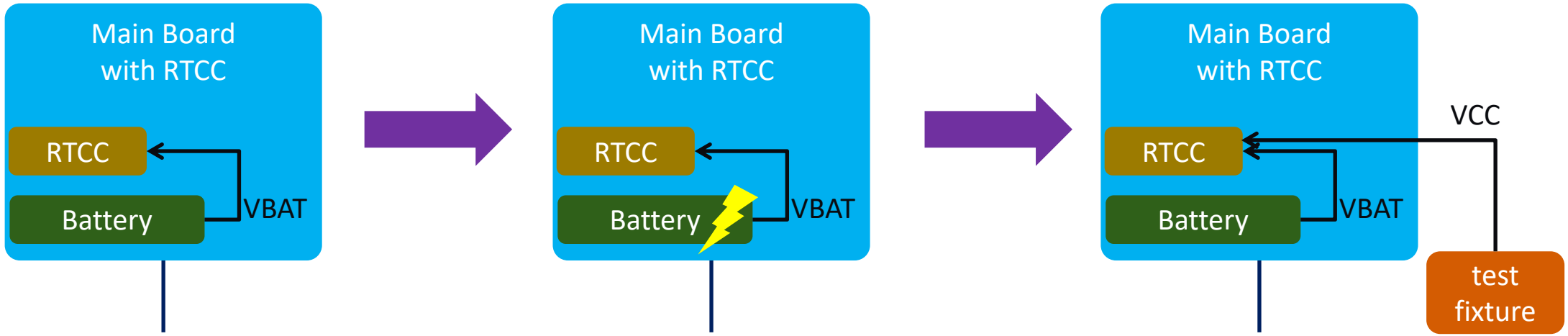
- With the failure chip on power unit, can we see the same situation after initializing the RTCC?

Sept-25, 2024

Reported by Weikeng Amond Lin

All tests were based on Weikeng’s test environment and conditions.

Production process with RTCC, without MCU.



Production Steps

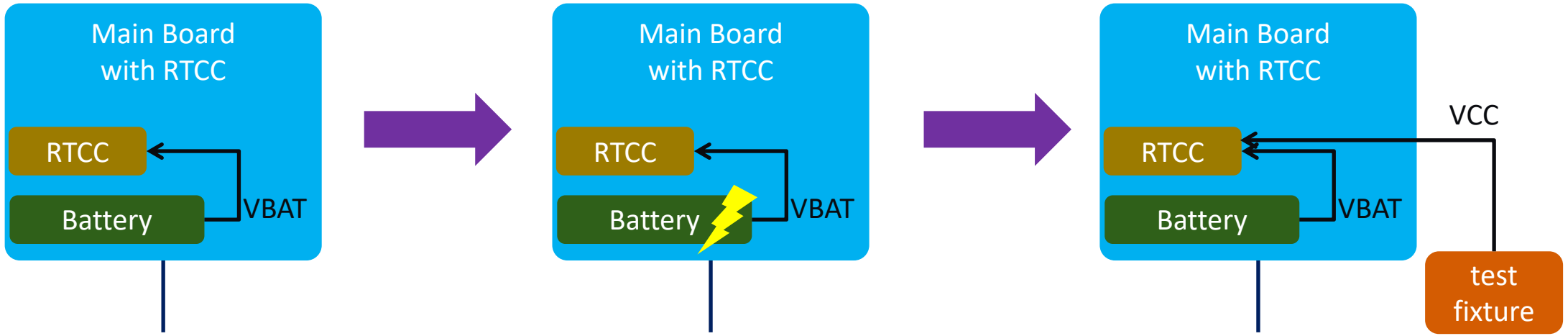
Step1. Battery assembly

Step2. Charging Station

Step3. RTCC Calibration Process

Final test process with RTCC & MCU.

*RTCC was never power-off.

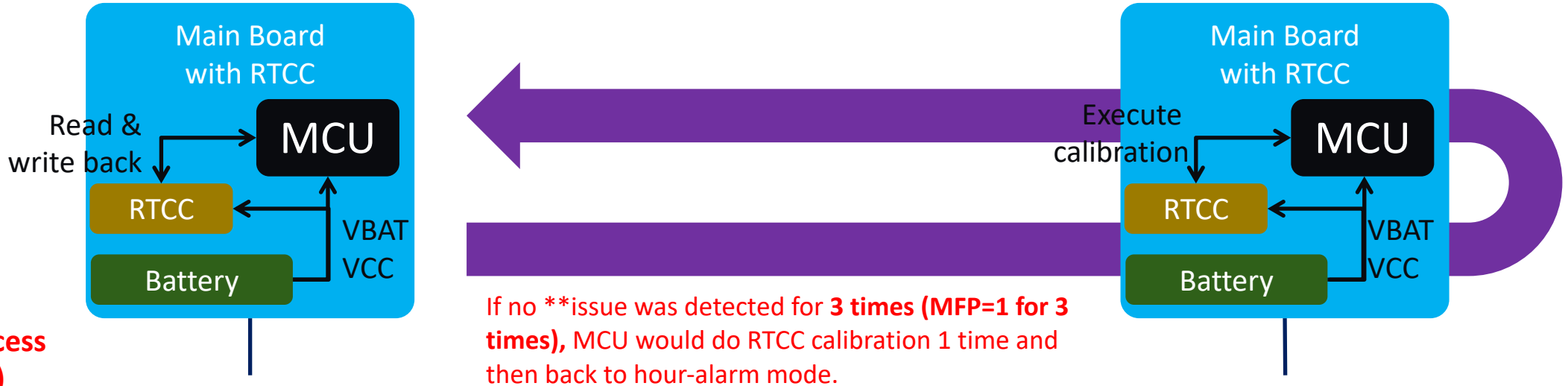


Production Steps

Step1. Battery assembly

Step2. Charging Station

Step3. RTCC Calibration Process



Final test Process (Normally)

Once MFP=1

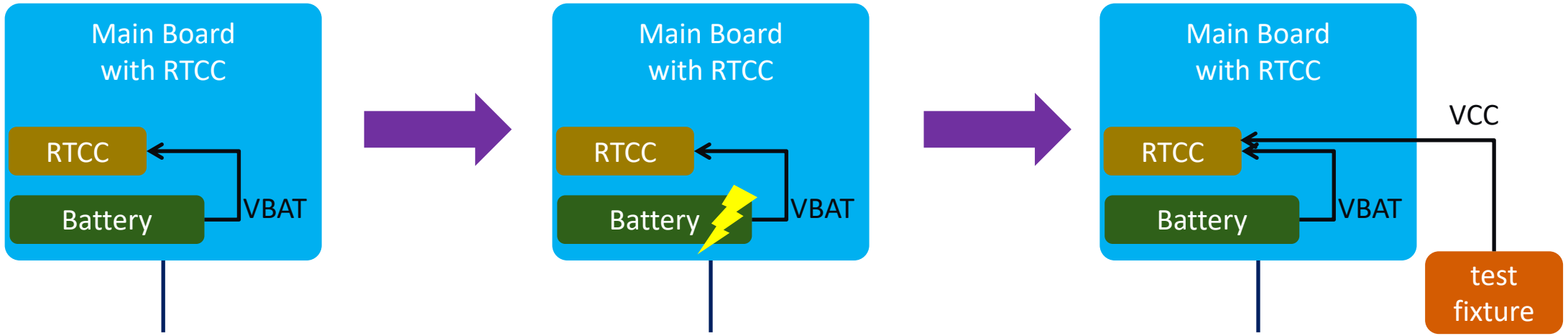
If no **issue was detected for **3 times (MFP=1 for 3 times),** MCU would do RTCC calibration 1 time and then back to hour-alarm mode.

**Issus means RTCHOUR != ALM0HOUR was detected but MFP=1.

RTCC Calibration Process

Final test process with RTCC & MCU.

*RTCC was never power-off.

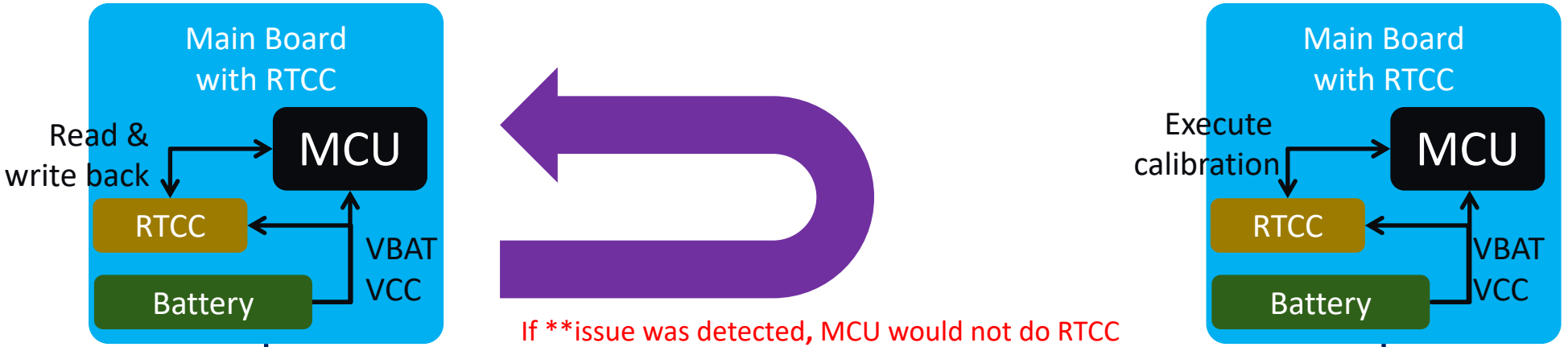


Production Steps

Step1. Battery assembly

Step2. Charging Station

Step3. RTCC Calibration Process



Final test Process (with **issue)

Once MFP=1

If **issue was detected, MCU would not do RTCC calibration and keep in hour-alarm mode.

RTCC Calibration Process

**Issus means RTCHOUR != ALM0HOUR was detected but MFP=1.

Question from BU to field

- **Question: If a chip fails as per the PPT (some even-numbered hours fail in 24 Hour mode), can a failed hour be repeated immediately? That is, if the fail at hour 22:00 is repeated again, does it fail again?**
  - Ans: Before RTCC is power-off, the issue would continue repeating and never stop. 22:00 was just one of examples. It could be any other hour. Additional update:
    - RTCC was always powered by battery and was never power-off.
    - The issue can be only reproduced with two conditions so far as we known:
      - Reproducible ONLY on client's product main board. (we have not been able to reproduce the issue on EVM.)
      - The failure chip can NOT be re-initialized again after "Production Steps". Re-initialization means the step3 of production steps. If did that again, the issue would not happen …at least we didn't see the issue again with the same failure chip and unit.

Question from BU to field

- **Question: Need to get some failed units to help confirm this issue.?**
  - Ans: Most failure units are still in factory. Will check with client for the failure units. On the other hands, Weikeng Amond has 1 failure chip but has not been able to reproduce the issue on our EVM.

Question from BU to field

- **Question: Need to get some failed units to help confirm this issue.?**
  - Ans: Most failure units are still in factory. Will check with client for the failure units. On the other hands, Weikeng Amond has 1 failure chip but has not been able to reproduce the issue on our EVM.

Suggestion from BU & Microchip Taiwan

- Depending on how the power is applied, this might be a problem. If the VBAT supply is applied first, then the VDD, then the registers may not initialize in the POR state. This means that some registers, like the CALIBRATION byte may not be as expected. This would likely vary from part to part.

- It is concerned about using the course TRIM setting to accelerate the testing. I don't know if that would be valid for testing alarms as we know there is already an errata regarding the TRIM value with the Minute alarm, so we will try to see if another method works better.
  - Feedback from field: It is true that we have not able to reproduce the issue with TRIM setting. We are not sure that accelerating the testing with TRIM setting is okay or not.

- It might be helpful if client can read TRIM back when the issue was happing?

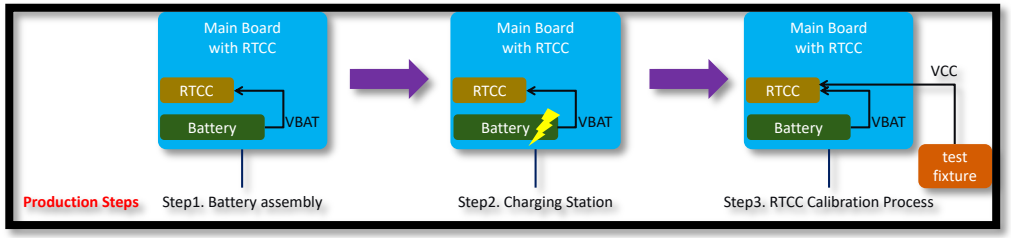
Sept-26, 2024

Reported by Weikeng Amond Lin

All tests were based on Weikeng’s test environment and conditions.

Question or Analysis from BU to field

- **The battery is applied first and it is charged. What is this, a rechargeable Lithium cell, or a SuperCap?**
  - Ans: This project is using rechargeable Lithium cell.



- **What is the charge voltage (slide 34)?**
  - Ans: Max 4V x 6 =24V battery package (6 cell in series)
  - VBAT would be from one cell of the 6 with max 4V.

- **We know that when the battery voltage is applied before VDD, that the registers do not have a known state. Until the VBATEN bit is turned on, there is no way that the device will store the registers correctly under battery power.**
  - Ans: Before calibration process, the registers do not have a known state. Step.3 (slide 34) is for this and for making sure VBATEN = 1 (process on slide 25&30).

- **The VDD looks like 3.3V per slide 19, also the I2C speed is 100kHz, which should be fine. I2C errors tend to be all or none, that is, if too fast it doesn't work at all but below that speed it always works. This can be variable when software I2C or bit-banging is used as it may not have a consistent timing like a hardware I2C port.**

Question or Analysis from BU to field

- **Slide 22, the VDD falling after MFP going high is OK, slide 23 VDD fall rate is fine, not too fast or too slow, exponential decay is typical for many systems. How long until the power is restored and what does the VDD reach just before it is turned on? Does it really reach 0V?**
  - Ans: Basically, MFP would be high for 23 hours until next hour-alarm event. So, VDD would have enough time to reach 0V (slide 24, start from 0V).
  - MFP would be high for at least 1 hours with issue. VDD still has enough time to reach 0V

- **Slide 24, VDD is powered on and MFP is low. Is this what is expected? If the MFP was high on power down, shouldn't it be high on power up? What changed here to explain this?**

- **Is the RTCC getting confused, such as if the VBATEN bit is not set? Slide 25 right side seems to show that VBATEN = 1, is this correct? If the bit is set, battery is applied and VDD falls and rises, then the MFP should be the same. Only the PowerFail bit should change and time should change.**
  - Ans: The VDD circuit would be enabled by MFP with active-low. But it is true that MFP could be possibly high by pull-up resistor before initializing RTCC (Step3, slide 34). After RTCC was initializd, MFP would be certain with active-low and normal high two mode. When MFP = high, power source of RTCC will be from battery only in power saving mode of unit, and it will be from both of battery and VDD in working mode of unit.

Question or Analysis from BU to field

- I want to clarify that the Course Trim function was never intended to be a normal operational mode, so it might be good to do testing of the alarms without it, perhaps by setting the time to within a few seconds before the alarm is expected. That would be something I would do if I had a unit that failed under the Course Trim

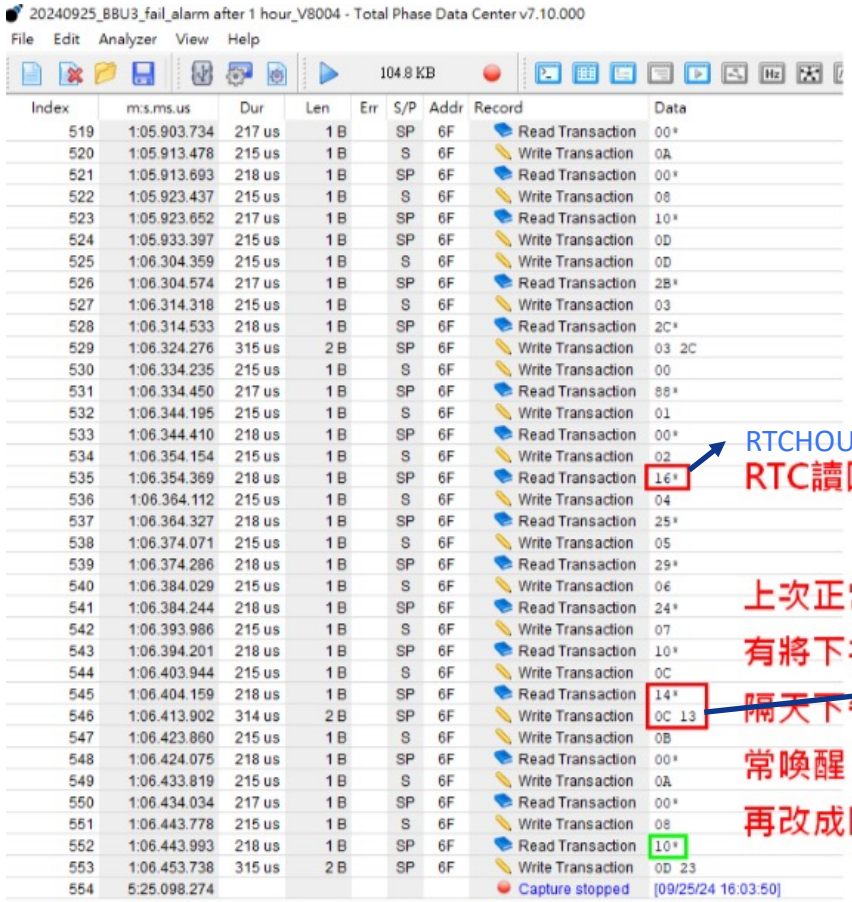
Feedback from client regarding OSCTRIM

- Read OSCTRIM (Never be initialized)

Unit #	Unit Status	Test with Source code #	OSCTRIM
SN001	Pass	V0905	144
SN005	Failure	V0905	73
SN006	Failure	V0905	87
SN007	Failure	V0905	80
SN008	Failure	V0905	7
SN009	Failure	V0905	34

Feedback from client regarding OSCTRIM

- Read OSCTRIM when issue was happening.



Index	m:s.ms.us	Dur	Len	Err	S/P	Addr	Record	Data
519	1:05.903.734	217 us	1 B		SP	6F	Read Transaction	00*
520	1:05.913.478	215 us	1 B		S	6F	Write Transaction	0A
521	1:05.913.693	218 us	1 B		SP	6F	Read Transaction	00*
522	1:05.923.437	215 us	1 B		S	6F	Write Transaction	08
523	1:05.923.652	217 us	1 B		SP	6F	Read Transaction	10*
524	1:05.933.397	215 us	1 B		SP	6F	Write Transaction	0D
525	1:06.304.359	215 us	1 B		S	6F	Write Transaction	0D
526	1:06.304.574	217 us	1 B		SP	6F	Read Transaction	2B*
527	1:06.314.318	215 us	1 B		S	6F	Write Transaction	03
528	1:06.314.533	218 us	1 B		SP	6F	Read Transaction	2C*
529	1:06.324.276	315 us	2 B		SP	6F	Write Transaction	03 2C
530	1:06.334.235	215 us	1 B		S	6F	Write Transaction	00
531	1:06.334.450	217 us	1 B		SP	6F	Read Transaction	88*
532	1:06.344.195	215 us	1 B		S	6F	Write Transaction	01
533	1:06.344.410	218 us	1 B		SP	6F	Read Transaction	00*
534	1:06.354.154	215 us	1 B		S	6F	Write Transaction	02
535	1:06.354.369	218 us	1 B		SP	6F	Read Transaction	16*
536	1:06.364.112	215 us	1 B		S	6F	Write Transaction	04
537	1:06.364.327	218 us	1 B		SP	6F	Read Transaction	25*
538	1:06.374.071	215 us	1 B		S	6F	Write Transaction	05
539	1:06.374.286	218 us	1 B		SP	6F	Read Transaction	29*
540	1:06.384.029	215 us	1 B		S	6F	Write Transaction	06
541	1:06.384.244	218 us	1 B		SP	6F	Read Transaction	24*
542	1:06.393.986	215 us	1 B		S	6F	Write Transaction	07
543	1:06.394.201	218 us	1 B		SP	6F	Read Transaction	10*
544	1:06.403.944	215 us	1 B		S	6F	Write Transaction	0C
545	1:06.404.159	218 us	1 B		SP	6F	Read Transaction	14*
546	1:06.413.902	314 us	2 B		SP	6F	Write Transaction	0C 13
547	1:06.423.860	215 us	1 B		S	6F	Write Transaction	0B
548	1:06.424.075	218 us	1 B		SP	6F	Read Transaction	00*
549	1:06.433.819	215 us	1 B		S	6F	Write Transaction	0A
550	1:06.434.034	217 us	1 B		SP	6F	Read Transaction	00*
551	1:06.443.778	215 us	1 B		S	6F	Write Transaction	08
552	1:06.443.993	218 us	1 B		SP	6F	Read Transaction	10*
553	1:06.453.738	315 us	2 B		SP	6F	Write Transaction	0D 23
554	5:25.098.274						Capture stopped	[09/25/24 16:03:50]

RTCHOUR was read back at 4pm.
RTC讀回時間為下午4點

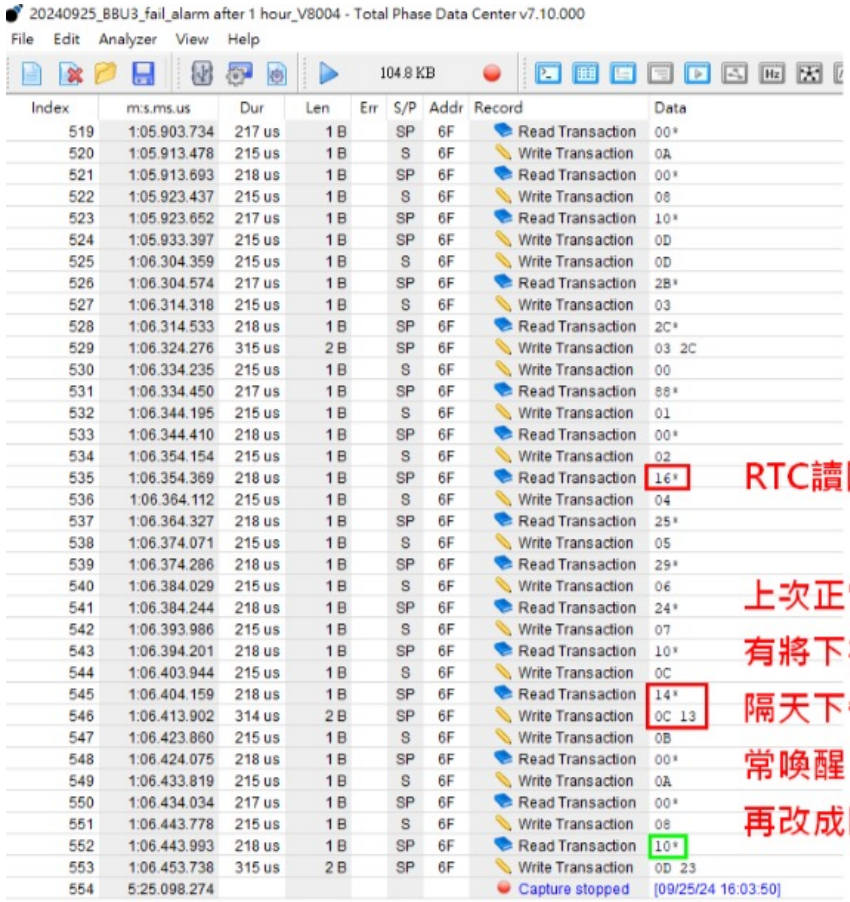
上次正常下午3點醒來，有將下次喚醒時間改成隔天下午2點，但這次異常喚醒，所以喚醒時間再改成隔天下午1點

Previous hour-alarm happened at 3pm properly, so then ALM0HOUR was configured to 2pm.
But because of the issue with one more time with hour-alarm, ALM0HOUR was configured to 1pm with decreasing one more hour.

讀回OSCTRIM暫存器，發現不是預設值0x00，而是0x10，但是不論校驗 or polling過程中，皆無對該暫存器寫入過，故此暫存器不應該有值

Question to BU

- Read OSCTRIM when issue was happening.


20240925_BBU3_fail_alarm after 1 hour_V8004 - Total Phase Data Center v7.10.000

File Edit Analyzer View Help

104.8 KB

Index	m:s.ms.us	Dur	Len	Err	S/P	Addr	Record	Data
519	1:05.903.734	217 us	1 B		SP	6F	Read Transaction	00*
520	1:05.913.478	215 us	1 B		S	6F	Write Transaction	0A
521	1:05.913.693	218 us	1 B		SP	6F	Read Transaction	00*
522	1:05.923.437	215 us	1 B		S	6F	Write Transaction	08
523	1:05.923.652	217 us	1 B		SP	6F	Read Transaction	10*
524	1:05.933.397	215 us	1 B		SP	6F	Write Transaction	0D
525	1:06.304.359	215 us	1 B		S	6F	Write Transaction	0D
526	1:06.304.574	217 us	1 B		SP	6F	Read Transaction	2B*
527	1:06.314.318	215 us	1 B		S	6F	Write Transaction	03
528	1:06.314.533	218 us	1 B		SP	6F	Read Transaction	2C*
529	1:06.324.276	315 us	2 B		SP	6F	Write Transaction	03 2C
530	1:06.334.235	215 us	1 B		S	6F	Write Transaction	00
531	1:06.334.450	217 us	1 B		SP	6F	Read Transaction	88*
532	1:06.344.195	215 us	1 B		S	6F	Write Transaction	01
533	1:06.344.410	218 us	1 B		SP	6F	Read Transaction	00*
534	1:06.354.154	215 us	1 B		S	6F	Write Transaction	02
535	1:06.354.369	218 us	1 B		SP	6F	Read Transaction	16*
536	1:06.364.112	215 us	1 B		S	6F	Write Transaction	04
537	1:06.364.327	218 us	1 B		SP	6F	Read Transaction	25*
538	1:06.374.071	215 us	1 B		S	6F	Write Transaction	05
539	1:06.374.286	218 us	1 B		SP	6F	Read Transaction	29*
540	1:06.384.029	215 us	1 B		S	6F	Write Transaction	06
541	1:06.384.244	218 us	1 B		SP	6F	Read Transaction	24*
542	1:06.393.986	215 us	1 B		S	6F	Write Transaction	07
543	1:06.394.201	218 us	1 B		SP	6F	Read Transaction	10*
544	1:06.403.944	215 us	1 B		S	6F	Write Transaction	0C
545	1:06.404.159	218 us	1 B		SP	6F	Read Transaction	14*
546	1:06.413.902	314 us	2 B		SP	6F	Write Transaction	0C 13
547	1:06.423.860	215 us	1 B		S	6F	Write Transaction	0B
548	1:06.424.075	218 us	1 B		SP	6F	Read Transaction	00*
549	1:06.433.819	215 us	1 B		S	6F	Write Transaction	0A
550	1:06.434.034	217 us	1 B		SP	6F	Read Transaction	00*
551	1:06.443.778	215 us	1 B		S	6F	Write Transaction	08
552	1:06.443.993	218 us	1 B		SP	6F	Read Transaction	10*
553	1:06.453.738	315 us	2 B		SP	6F	Write Transaction	0D 23
554	5:25.098.274						Capture stopped	[09/25/24 16:03:50]

RTC讀回時間為下午4點

上次正常下午3點醒來，有將下次喚醒時間改成隔天下午2點，但這次異常喚醒，所以喚醒時間再改成隔天下午1點

讀回OSCTRIM暫存器，發現不是預設值0x00，而是0x10，但是不論校驗 or polling過程中，皆無對該暫存器寫入過，故此暫存器不應該有值

Client expected the OSCTRIM should be kept with 0x00 because OSCTRIM had never been initialized. But it was 0x10 this time.

Recommend:

The register initialization is likely a problem. It appears that the value in OSCTRIM is not as expected, and this happens when VBAT is applied to RTCC before VDD, so the POR reset of all registers does not happen. This suggests that on initial test that all the registers be initialized in code to ensure that they don't have unexpected values. This would be easier than changing the power sequencing.

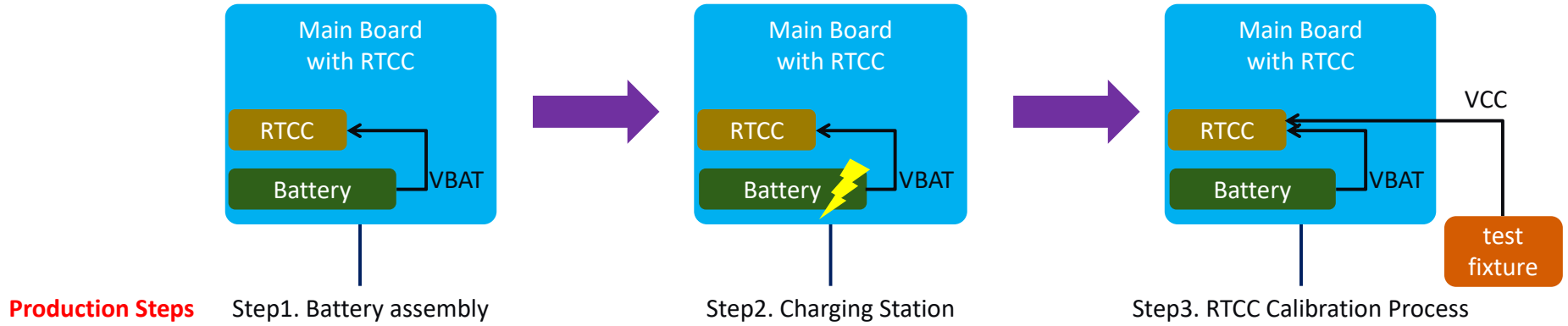
Oct-1, 2024

Reported by Weikeng Amond Lin

All tests were based on Weikeng’s test environment and conditions.

Co-worked with AcBel to confirm testing process

- **Calibration flowchart checking: OK (slide 30).**
  - We coworked to check Weikeng's test code included polling RTCC and initialization. All the process including calibration flowchart matched R&D's requirement.
- **The power-up sequence checking: OK, VCC applied after VBAT**
  - On AcBel production line, during assembly (step1), VBAT will be applied first, and VCC will not be applied until the unit moves to step3, RTCC calibration process.

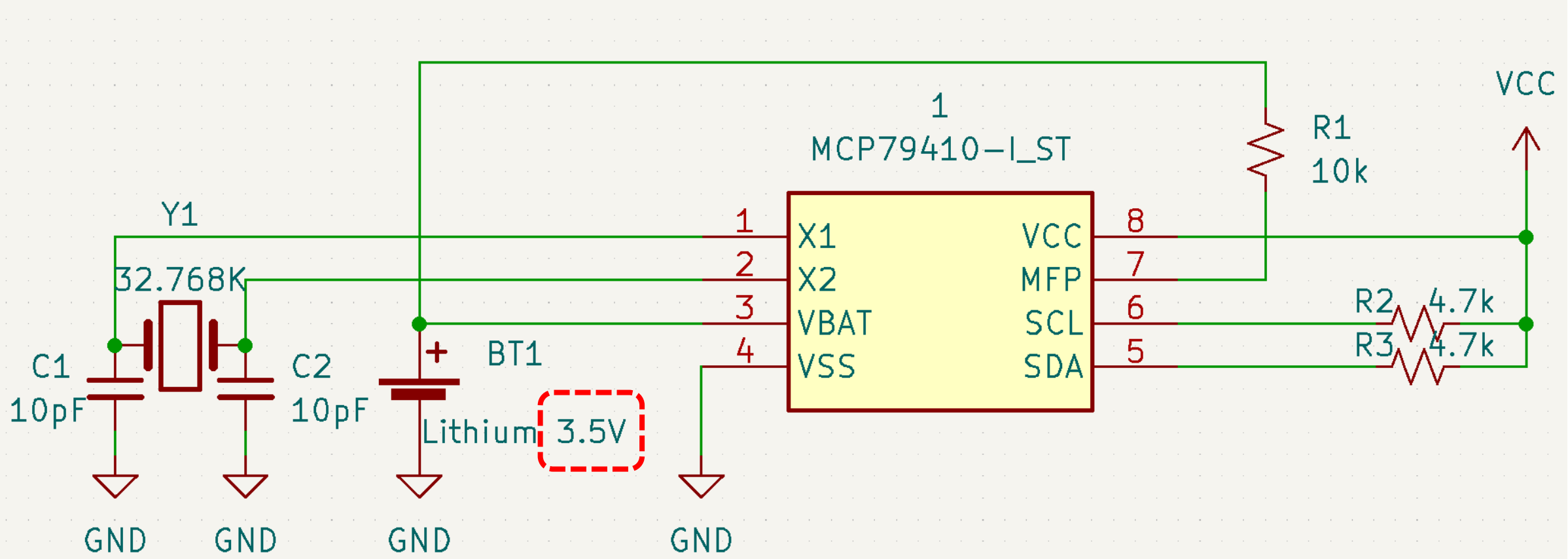


  - With the power sequencing manually, OSCTRIM register was always 0x00 (read on Weikeng's test platform). It seems also not easy to reproduce "Power-on reset" issue as well. OSCTRIM register was 0x00 or 0xFF (read on AcBel's test platform). It was uncertain (showed on slide45 as well) whether this difference has any impact.

Next Action

- **Remove VCC, VBAT and follow the power sequence.**
  - According to the power sequence, AcBel can observe that the OSCTRIM register is either 0x00 or 0xFF. Therefore, during initialization, if MCU found OSCTRIM was not 0x00, MCU will write 0x00 to OSCTRIM. This process will be used to test whether the issue can be reproduced.
- **Weikeng will implement a new testing method**
  - R&D mentioned that there is a special process which he modified to reproduce the issue. Not sure it is necessary or not, but it is recommended to do the same way.
    - Set the 1st alarm for the next hour, then set another alarm for the next 23 hours when 1st alarm occured. The purpose is to observe the same situation on the Weikeng test platform compared with AcBel platform. It would be easier to see issue after 1~3 hours (or took more time to see the issue.)

Weikeng test platform circuit



WEIKENG

MICROCHIP

Thanks.