Ace2 Customer OTP Programming Flow

v2, Will Ferry, 05/24/2019

Connections required:
- System DCIN (Must be able to provide ≥12V to this rail externally on Notebooks, assumes Desktops connected PP_HV to 12V rail internally)
- Ace2 I2C2_SCL/SDA (NOTE: This bus is 1.8V, I2C master must support 1.8V)
- GND

First Ace2 7-bit I2C address is **0x38**.
Second Ace2 7-bit I2C address is **0x3F**.

If four Aces:
Third Ace2 7-bit I2C address is **0x20**.
Fourth Ace2 7-bit I2C address is **0x27**.

NOTE: Some specific systems may use additional or different addresses, or I2C addresses may be changed once OTP is programmed. Details will be provided with system-specific OTP.bin files if necessary.

The Ace2 customer-accessible OTP region contains 3 chunks of data:
1) "Customer Words": 8 bytes of data that configures boot behavior
2) "Application Customization": Up to 1156 bytes of data that configures normal behavior
3) "Key Data": TBD bytes used for secure FW updates (more data on this to be provided later)
NOTE: Customer Words will be different between development systems and revenuable systems, different files will be provided.

Ace2 OTP.bin file format (v2):

Unlike Ace, which only had a pair of 32-bit values to be programmed, Ace2 has more data and each location in a system will require unique programming, so a file will be provided containing the data to be programmed into each Ace2 location.

This file is defined as 1308 bytes (v2). The first 1280 bytes represent the full Ace2 OTP memory region, though not all of the data in this region is customer-accessible. The final 28 bytes are 7 additional 32-bit (little-endian) values provided for the programming flow. More details provided for each below:

Word0: Previous CRC
Word1: Application Customization data start
Word2: Application Customization data size
Word3: Key Data start
Word4: Key Data size (lower 8 bits) / Key Data flags (upper 8 bits)
Word5: Final CRC
Word6: Key Data CRC
NOTE: Older versions of OTP.bin may be 1304 bytes (v1) and lack Word6, Key Data programming is NOT supported on those images, skip steps 7-9.

Previous CRC:
- Since Ace2 has multiple items stored in the OTP, it is possible to flash each of these items separately. In addition, Application Customization data can be appended by storing multiple blocks. So this file format allows for both Ace2 parts where the Customer Words (only) have previously been programmed, or there may be existing Application Customization data with more data being written by the new OTP.bin file. The CRC provided here verifies that the existing OTP matches expectations, if the CRC returned by Ace2 does not match the provided value (or the special value 0xD42C0C06, which represents an empty OTP region) then flashing cannot proceed.

Application Customization data start/size:
- When this size field is non-zero, the OTP.bin contains Application Customization data. This data is provided to Ace2 during a certain (optional) part of the OTP programming flow. These fields inform the programmer where this data starts in the OTP.bin file and how many bytes are to be read from the file during this phase.

Key Data start/size/flags/CRC:
- When this size field is non-zero, the OTP.bin contains Key Data. This data is provided to Ace2 during a certain (optional) part of the OTP programming flow. These fields inform the programmer where this data starts in the OTP.bin file and how many bytes are to be read from the file during this phase.

NOTE: Unlike Ace, there should be no requirement on the time between I2C transactions for Ace2. The programmer should consider still supporting the Ace I2C timing requirements (each I2C transaction performed within 90ms of the previous one, first transaction may be NAK'ed, if this happens repeat the failed transaction with a delay of >15ms but <90ms.

Procedure (entire flow is to be performed for each Ace2, independently):
1) Power-up board (DCIN/12V and G3Hot), hold SMC in RESET# (only G3Hot power is required for Ace functionality, all other rails should be kept off). SMC must be in RESET# (or DFU mode) or it may interfere with I2C communications to Ace.

2) Verify Ace2 operating mode.
    a) Perform I2C read (5 bytes) to Ace2 subaddress 0x03.
    b) Response must be **EITHER** 0x04, 0x41, 0x50, 0x50, 0x20 **OR** 0x04, 0x44, 0x46, 0x55, 0x66 **OR** 0x04, 0x42, 0x4F, 0x4F, 0x54, otherwise abort (can attempt power-cycle but results not likely to be different).

3) Verify current Customer-Accessible OTP region state.
    a) Perform I2C write (2 bytes) to Ace2 subaddress 0x11, data is 0x01 followed by 0x80.
    b) Perform I2C write (5 bytes) to Ace2 subaddress 0x10, data is 0x04 followed by 0x4F, 0x54, 0x50, 0x72.
    c) Perform I2C read (5 bytes) from Ace subaddress 0x10. Should read back one of two different values, either 0x04, 0x4F, 0x54, 0x50, 0x72 or 0x04, 0x00, 0x00, 0x00, 0x00. If the first value is encountered, repeat this step. If the second value is encountered continue to the next step. If any other value is encountered abort (part failed to read OTP region, unexpected error).
    d) Perform I2C read (5 bytes) from Ace subaddress 0x11. First byte will be 0x40, remaining 4 bytes will be CRC of current OTP region (little-endian). If value is 0x06, 0x0C, 0x2C, 0xD4 (0xD42C0C06) then OTP region is empty, all data must be programmed. If value matches "Previous CRC" provided in OTP.bin file then only Application Customization and/or Key Data must be written, as per size fields in OTP.bin file. If value matches "Final CRC" provided in OTP.bin then this Ace2 has already been programmed with provided data, skip all remaining steps.

Steps 4-6 describe the Application Customization-specific part of the flow. These steps should be skipped if Application Customization data size == 0.
4) Prepare Ace2 for receipt of Application Customization data.
    a) Perform I2C write (2 bytes) to Ace2 subaddress 0x11, data is 0x01 followed by 0x00.
    b) Perform I2C write (5 bytes) to Ace2 subaddress 0x10, data is 0x04 followed by 0x4F, 0x54, 0x50, 0x69.
    c) Perform I2C read (5 bytes) from Ace subaddress 0x10. Should read back one of two different values, either 0x04, 0x4F, 0x54, 0x50, 0x69 or 0x04, 0x00, 0x00, 0x00, 0x00. If the first value is encountered, repeat this step. If the second value is encountered continue to the next step. If any other value is encountered abort (part failed to initialize OTP buffer, unexpected error).
    d) Perform I2C read (5 bytes) from Ace subaddress 0x11. First byte will be 0x40. Second byte must be 0x00 (otherwise an error occurred, abort). Third byte is ignored. Last two bytes (little-endian) indicate remaining space for Application Customization data. If this value is less than Application Customization data size field then abort (shouldn't happen).

Step 5 is repeated for every 64 bytes ("block") of Application Customization data.
5) Provide block of Application Customization data to Ace2.
    a) Perform I2C write (65 bytes) to Ace2 subaddress 0x11, data is 0x40 followed by 64 bytes from OTP.bin, in order provided in OTP.bin starting at (zero-indexed) address in Application Customization data start field. For final block if less than 64 bytes pad remaining bytes with 0x00.
    b) Perform I2C write (5 bytes) to Ace2 subaddress 0x10, data is 0x04 followed by 0x4F, 0x54, 0x50, 0x64.
    c) Perform I2C read (5 bytes) from Ace subaddress 0x10. Should read back one of two different values, either 0x04, 0x4F, 0x54, 0x50, 0x64 or 0x04, 0x00, 0x00, 0x00, 0x00. If the first value is encountered, repeat this step. If the second value is encountered continue to the next step. If any other value is encountered abort (part failed to store OTP data, unexpected error).
    d) Perform I2C read (3 bytes) from Ace subaddress 0x11. First byte will be 0x40. Second byte must be 0x80

for all blocks except for the final block, and 0x00 for the final block (otherwise an error occurred, abort). Third byte represents next block number, so first iteration of step 5 should return 1, second will return 2, etc. If unexpected value is returned then there has been a sequence error, abort (can retry by starting at step 4).

6) Complete (write) Application Customization data.
a) Perform I2C write (2 bytes) to Ace2 subaddress 0x11, data is 0x40 followed by 0x02.
b) Perform I2C write (5 bytes) to Ace2 subaddress 0x10, data is 0x04 followed by 0x4F, 0x54, 0x50, 0x77.
c) Perform I2C read (5 bytes) from Ace subaddress 0x10. Should read back one of two different values, either 0x04, 0x4F, 0x54, 0x50, 0x77 or 0x04, 0x00, 0x00, 0x00, 0x00. If the first value is encountered, repeat this step. If the second value is encountered continue to the next step. If any other value is encountered abort (part failed to write OTP region, unexpected error).
d) Perform I2C read (2 bytes) from Ace subaddress 0x11. First byte will be 0x40. Second byte must be 0x00 (otherwise an error occurred, abort).

Steps 7-9 describe the Key Data-specific part of the flow. These steps should be skipped if Key Data size == 0.
7) Prepare Ace2 for receipt of Key Data.
a) Perform I2C write (2 bytes) to Ace2 subaddress 0x11, data is 0x01 followed by (Key Data flags, should be 0x01, 0x02 or 0x03).
b) Perform I2C write (5 bytes) to Ace2 subaddress 0x10, data is 0x04 followed by 0x4F, 0x54, 0x50, 0x69.
c) Perform I2C read (5 bytes) from Ace subaddress 0x10. Should read back one of two different values, either 0x04, 0x4F, 0x54, 0x50, 0x69 or 0x04, 0x00, 0x00, 0x00, 0x00. If the first value is encountered, repeat this step. If the second value is encountered continue to the next step. If any other value is encountered abort (part failed to initialize OTP buffer, unexpected error).
d) Perform I2C read (4 bytes) from Ace subaddress 0x11. First byte will be 0x40. Second byte must be 0x00 (otherwise an error occurred, abort). Third byte is ignored. Last byte indicates remaining space for Key Data. If this value does not match Key Data size field then abort (shouldn't happen).

8) Provide block of Key Data data to Ace2.
a) Perform I2C write (65 bytes) to Ace2 subaddress 0x11, data is 0x40 followed by (Key Data size, should be 64 or 32) bytes from OTP.bin, in order provided in OTP.bin starting at (zero-indexed) address in Key Data start field. If Key Data size == 32 add 32 bytes of 0x00 to write data (always write 65 bytes total).
b) Perform I2C write (5 bytes) to Ace2 subaddress 0x10, data is 0x04 followed by 0x4F, 0x54, 0x50, 0x64.
c) Perform I2C read (5 bytes) from Ace subaddress 0x10. Should read back one of two different values, either 0x04, 0x4F, 0x54, 0x50, 0x64 or 0x04, 0x00, 0x00, 0x00, 0x00. If the first value is encountered, repeat this step. If the second value is encountered continue to the next step. If any other value is encountered abort (part failed to store OTP data, unexpected error).
d) Perform I2C read (2 bytes) from Ace subaddress 0x11. First byte will be 0x40. Second byte must be 0x00 (otherwise an error occurred, abort).

9) Complete (write) Key Data.
a) Perform I2C write (14 bytes) to Ace2 subaddress 0x11, data is 0x40 followed by 0x04, 8 bytes of 0x00, and Key Data CRC (4 bytes, little-endian).
b) Perform I2C write (5 bytes) to Ace2 subaddress 0x10, data is 0x04 followed by 0x4F, 0x54, 0x50, 0x77.
c) Perform I2C read (5 bytes) from Ace subaddress 0x10. Should read back one of two different values, either 0x04, 0x4F, 0x54, 0x50, 0x77 or 0x04, 0x00, 0x00, 0x00, 0x00. If the first value is encountered, repeat this step. If the second value is encountered continue to the next step. If any other value is encountered abort (part failed to write OTP region, unexpected error).
d) Perform I2C read (2 bytes) from Ace subaddress 0x11. First byte will be 0x40. Second byte must be 0x00 (otherwise an error occurred, abort).

Step 10 describes the Customer Words-specific part of the flow. This step should be skipped if CRC from step 3 matched Previous CRC in OTP.bin (indicating the Customer Words are already programmed).
10) Write Customer Words data.
a) Perform I2C write (10 bytes) to Ace2 subaddress 0x11, data is 0x40 followed by 0x01 and first 8 bytes of OTP.bin file in order.
b) Perform I2C write (5 bytes) to Ace2 subaddress 0x10, data is 0x04 followed by 0x4F, 0x54, 0x50, 0x77.
c) Perform I2C read (5 bytes) from Ace subaddress 0x10. Should read back one of two different values, either 0x04, 0x4F, 0x54, 0x50, 0x77 or 0x04, 0x00, 0x00, 0x00, 0x00. If the first value is encountered, repeat this step.

If the second value is encountered continue to the next step. If any other value is encountered abort (part failed to write OTP region, unexpected error).

d) Perform I2C read (2 bytes) from Ace subaddress 0x11. First byte will be 0x40. Second byte must be 0x00 (otherwise an error occurred, abort).

11) Verify final Customer-Accessible OTP region state.

a) Perform I2C write (2 bytes) to Ace2 subaddress 0x11, data is 0x01 followed by 0x80.

b) Perform I2C write (5 bytes) to Ace2 subaddress 0x10, data is 0x04 followed by 0x4F, 0x54, 0x50, 0x72.

c) Perform I2C read (5 bytes) from Ace subaddress 0x10. Should read back one of two different values, either 0x04, 0x4F, 0x54, 0x50, 0x72 or 0x04, 0x00, 0x00, 0x00, 0x00. If the first value is encountered, repeat this step. If the second value is encountered continue to the next step. If any other value is encountered abort (part failed to read OTP region, unexpected error).

d) Perform I2C read (5 bytes) from Ace subaddress 0x11. First byte will be 0x40, remaining 4 bytes will be CRC of OTP region (little-endian). If value does not match "Final CRC" provided in OTP.bin then programming has failed (should have failed before reaching this step), abort.