BitCloud SDK 3.2.0 Migration Guide

1 Introduction

BitCloud SDK v3.2.0 combines Atmel ZigBee PRO based applications for ZigBee Light Link, ZigBee Home Automation and vendor-specific profiles. It introduces updated applications and improved core stack compared to previous public releases of BitCloud 3.1.0.

This document describes key changes in the BitCloud stack and applications code that might require porting efforts for custom applications developed on previous versions of BitCloud SDK.

2 Changes in BitCloud stack components that impact application

1. EEPROM Emulation on SAMR21: Implementation specific, the last 4KB of Flash memory in SAMR21 is used for EEPROM Storage in general. Since the emulation functionality was not available in the previous release, the halFlashWriteEepromPage() function when called takes a copy of the data in RAM and writes new data to the flash/eeprom. This would be slightly dangerous and vulnerable to resets and power drops during the write to EEPROM, hence causing loss of data (selected page). To overcome the issue, EEPROM emulation feature has been added as part of the current release and it is protected by a switch named `EEPROM_EMULATION` as part of `HAL Configuration.` The eeprom emulation size can be changed to different size based on the need. For more information please refer fuse bits `NVMCTRL_EEPROM_SIZE` from Atmel Studio IDE after connecting the SAMR21 device and its corresponding memory allocation in SAMR21 datasheet, i.e., `"Table 20-3. Flash size for EEPROM emulation".` The changes done for the eeprom emulation makes the bootloader not backward compatible as in the previous release we had a constant memory region for the same. This implies that when `EEPROM_EMULATION` is enabled(by default it is disabled) the user cannot use older version of BitCloud applications with the current release of bootloader and older release of bootloader with current release of applications.(BCZPRO-1456)

2. Multiple definitions of Manufacturer code got cleaned up. The definition of `CS_MANUFACTURER_CODE` is moved to application `configuration.h` file for ZLL and HA.(BCZPRO-421)

3. With the current release the ZCL would be able to handle full-size frames i.e., 127 bytes at PHY layer. The APS_MAX_ASDU_SIZE for various security configurations is as mentioned in BitCloud Developers Guide. The OFD_BLOCK_SIZE is now defined as `#define OFD_BLOCK_SIZE (APS_MAX_NON_SECURITY_ASDU_SIZE- ZCL_IMAGE_BLOCK_RESP_HEADER_SIZE - ZCL_FRAME_STANDARD_HEADER_SIZE)` (BCZPRO-1188)

4. OTA has been redesigned in the current release and the list of changes w.r.t to files/functionalities are listed below (BCZPRO-436),
   I. `zclOtauDiscovery.c and zclOtauDiscovery.c` does not exist anymore in the release.
   II. `zclOtauClientDiscovery.c, zclOtauClientDownload.c , zclOtauClientPdt.c, zclOtauClientQuery.c and zclOtauClientUpgrade.c` and corresponding header files are newly added
   III. Makefiles for both IAR and GCC for ZLL and ZHA, WSN Demo updated with the new file additions and old file deletion. The details are given in `Makefile_All_StdlinkSec_SamR21_Atsamr21g18a_Rf233_48MHz_Iar (`For eg.) which is located at ..`\Applications\ZLLDemo\makefiles\ATSAMR21G18A\..`
   IV. A new definition `APP_SUPPORT_OTAU_RECOVERY` is added as part of application configuration to support `OTAU` recovery mechanism
   V. A configurable time parameter for querying server for `OTAU` clients is supported by a macro `CS_ZCL_OTAU_QUERY_INTERVAL` in application configuration.
   VI. Maximum retry counters for `OTAU` client is also configurable by a macro defined as `CS_ZCL_OTAU_MAX_RETRY_COUNT` in application configuration.

5. The existing behavior of end device running on 32-Khz crystal and while sleeping, rejoins to its parent each time an external interrupt is given got changed to not to rejoin. Changes are done in files `colorSceneRemote.c , dimmerSwitch.c and multiSensor.c` and the data supplied to the `HAL_EVENT_WAKING_UP` event handler is modified in the function `sleepEventHandler().` (BCZRO-1617)

6. All the other major features/enhancements/bug fixes are already captured as part of the release notes.

3 Changes in BitCloud reference applications

This section describes changes in BitCloud reference applications (delivered in `Applications/` folder of BitCloud SDK). Some of the modifications are related to the changes in the stack components described in Chapter 2, while others are application-specific.

3.1 Common for all applications

7. Reference application projects are ported to new versions of Atmel Studio and IAR IDE. See `Documentation/AVR2052_BitCloud_SDK_Quick_Start_Guide.pdf` for actual versions. Also application structure, linker scripts and Make files have been updated.

8. User Button in SAMR21-XPRO and ATmega256RFR2-XPRO used for resetToFN functionality works as expected . Updates are done in ZLL Demo files `colorSceneRemote.c` ,light.c by checking/adding support for the mentioned(SAMR21-XPRO/ ATmega256RFR2-XPRO) `BSP` in `init` and e`nableResetToFNButton` function respectively.(BCZPRO-1079)

3.2 ZLLDemo-specific

ZLLDemo application code has been updated to support new BitCloud features and -bug-fixes. It is strongly recommended to reuse updated ZLLDemo application code for custom ZLL applications. This section lists all the ZLL-specific modifications applied in addition to those given in Section 2 and Section 3.1.

9. A ZLL Remote which was earlier continuously polling even after parent loss now sets the sleep period to zero on the disconnected event and restores the same(sleep period) on next connected state i.e., finding the next suitable parent. The Disconnected and Connected function in `colorSceneRemoteFsm.c` got updated for the change by storing and restoring the `sleepPeriodConfigured` value during disconnect and connect state respectively, (BCZPRO-1239)

10. After the device loses its parent and goes into network discovery, it retains the extended PANID information and is able to scan and join to a parent on other channels set in primary and secondary channel masks. Updates are done in file `colorSceneRemote.c` and `initMacLayerDone` function,(BCZPRO-1514)

11. ZLL Add Group command corrected to take group ID from a given range Updates are done in function `processAddGroupCmd` in files `bridgeConsole.c, colorSceneRemoteConsole.c, SYS_GetRandomNumber`() fcuntion call within SendAddGroup function in file `colorSceneRemoteFsm.c` replaced by a new function `N_AddressManager_AllocateGroupId(&appData.groupId)` to get the valid group ID range.(BCZPRO-1513)

12. The behavior of the ZLL application changed when the device receives a Network Leave command over the air now. The device would do a reset to factory new and changes for this are done in `light.c`, by subscribing to `BC_EVENT_LEAVE_COMMAND_RECEIVED` event in `initMacLayerDone` and corresponding actions within the function. In `colorSceneRemoteFsm.c` file and `ColorSceneRemote_FsmInit` function, `N_Connection_Callback_t` structure is updated with an addition of `ResetToFactoryDefaults` and its related actions.(BCZPRO-1472)

13. Immediately after Touchlink between Remote and Light, when we reset Light and initiate ON/OFF command, the LED behavior is changed as expected. Changes are done in.`colorControlClusterInit` function in `lightColorControlCluster.c, levelControlClusterInit` function in `lightLevelControlCluster.c , onOffClusterInit` function in `onOffCluster.c, scenesClusterInit` function in `lightScenesCluster.c` , check for `if( N_DeviceInfo_IsFactoryNew() )` is replaced by if(`!PDS_IsAbleToRestore(APP_LIGHT_COLOR_CONTROL_CLUSTER_SERVER_ATTR_MEM_ID)), If(!PDS_IsAbleToRestore(APP_LIGHT_LEVEL_CONTROL_CLUSTER_SERVER_ATTR_MEM_ID)), If(!PDS_IsAbleToRestore(APP_LIGHT_ONOFF_CLUSTER_SERVER_ATTR_MEM_ID)), If(!PDS_IsAbleToRestore(APP_LIGHT_SCENE_CLUSTER_SERVER_ATTR_MEM_ID)),respectively.` (BCZPRO-1235),

14. Instances of CLSN-Pro profile and its associated functionalities completely removed from ZLL Application. Changes done in files and functions are as follows, (BCZPRO-1524)

    I. N_DeviceInfo_Init function has changed because of the removal of obsolete functionality and a new API can be found in N_DeviceInfo_Init.h. The function

    ```
    N_DeviceInfo_Init(N_DeviceInfo_Profile_t profile,
    uint8_t touchlinkRssiCorrection, int8_t touchlinkRssiThreshold, int8_t
    buttonlinkRssiThreshold , bool addressAssignmentCapable, bool
    canBeLinkInitiator, uint8_t touchlinkZerodBmTransmitPower)
    ```

    Changed to

    ```
    N_DeviceInfo_Init(uint8_t touchlinkRssiCorrection, int8_t
    touchlinkRssiThreshold,bool addressAssignmentCapable, bool
    canBeLinkInitiator, uint8_t touchlinkZerodBmTransmitPower)
    ```

    The new definition of N_DeviceInfo_Init is replaced with the existing in bridge.c, colorSceneRemote.c, light.c.

    II. N_DeviceInfo_Profile_t structure updated in N_DeviceInfo.h by removing N_DeviceInfo_Profile_CLSN_Pro element.

    III. N_DeviceInfo_GetProfile_Impl and N_DeviceInfo_GetButtonlinkRssiThreshold_Imp functionality removed from N_DeviceInfo.c and its usage across files in ZLL.

    IV. N_LinkInitiator_ReverseJoinDevice_Impl removed from colorSceneRemoteBindings.h file. ReceivedChallengeResponse and ReceivedNetworkReverseJoinRequest

    V. EVENT_REVERSE_JOIN_INDICATION, removed from colorSceneRemoteFsm.c file and all associated functionalities w.r.t the event no more exists.

15. Recall scene command for more than one scene which was not working previously, fixed as expected. The changes are done in colorControlClusterSetExtensionField function in lightColorControlCluster.c by modifying the existing arguments passed to the function and calling this function within recallScene function in lightScenesCluster.c (BCZPRO-1648)

16. Get Group Membership command which was earlier returning 5 (MAX_GROUPS_IN_GROUP_MEMBERSHIP_RESPONSE) groups in the response regardless of the actual number, got fixed as part of the release. Updates are done in files lightGroupsCluster.c (getGroupMembershipInd function) and zclZllGroupsCluster.h (ZCL_GetGroupMembershipResponse_t structure) by changing the MAX_GROUPS_IN_GROUP_MEMBERSHIP_RESPONSE macro to CS_GROUP_TABLE_SIZE (BCZPRO-1157)

17. ZLL Remote which was failing to rejoin to its parent when it loses its parent got fixed , the change is done in file networkJoining.c and in assocJoinDone function by checking csExtPanId is non zero value or not. After joining any network the csExtPanId shall have a value other than zero.(BCZPRO-1687)

3.3 HADevice-specific

HADevice reference application has been modified in its structure and functionality. New device types are added and functionality updated as per ZHA 1.2 specification. For details see application description in Documentation/AVR2052_BitCloud_SDK_Quick_Start_Guide.pdf. This section lists all the ZHA-specific modifications applied in addition to those given in Section 2 and Section 3.1.

18. Handling of configure reporting in MultiSensor has been modified i.e., the consecutive `sendConfigureReportingToNotify` has been changed to individual `sendConfigureReportingToNotify` after receiving the confirmation callback. Updates are done in files `dimmableLight.c , multiSensor.c` (BCZPRO-1774, BCZPRO-1716)

19. Combined Interface can now simultaneously act as both Initiator and target for HA EZ-Mode.
    If two devices (CI and MultiSensor) are in a network and both being initiators, the effect of users having to wait for 3 minutes for the EZ-mode commissioning to complete and then set explicitly CI as target device for Ez-mode commissioning is avoided by this implementation. Update done in `zdoPermitJoiningResponse` function in `ezModeManager.c` file.(BCZPRO-1270)

20. Combined Interface is updated with the CIE (Control and Indicating Equipment) functionalities to work with IAS ACE ..Updates are done in the following files w.r.t to the new device type addition. (BCZPRO-1563)

    I. The `ConsoleCommand_t cmds`[] in `ciConsole.c` got updated with the new command addition for IAS ACE
    II. `ZCL_Cluster_t ciClientClusters[CI_CLIENT_CLUSTERS_COUNT]` and `ZCL_Cluster_t ciServerClusters[CI_SERVER_CLUSTERS_COUNT`] got updated with IAS ACE clusters addition.
    III. `CI_SERVER_CLUSTERS_COUNT` and `CI_CLIENT_CLUSTERS_COUNT` got updated in `ciClusters.h`
    IV. New device type addition updated in `appConsts.h` as `APP_DEVICE_TYPE_IAS_ACE`
    V. Since new command set added the `APP_UART_TX_BUFFER_SIZE` definition in `uartManager.c` got updated to a higher value.

21. Combined Interface is updated to work with Thermostat. Updates are done in the following files w.r.t to the new device type addition (BCZPRO-1232)

    I. The `ConsoleCommand_t cmds[]` in `ciConsole.c` got updated with the new command addition for IAS ACE
    II. `ZCL_Cluster_t ciClientClusters[CI_CLIENT_CLUSTERS_COUNT]` and `ZCL_Cluster_t ciServerClusters[CI_SERVER_CLUSTERS_COUNT]` got updated with IAS ACE clusters addition.
    III. `CI_SERVER_CLUSTERS_COUNT` and `CI_CLIENT_CLUSTERS_COUNT` got updated in `ciClusters.h`
    IV. New device type addition updated in `appConsts.h` as `APP_DEVICE_TYPE_THERMOSTAT`
    V. Since new command set added the `APP_UART_TX_BUFFER_SIZE` definition in `uartManager.c` got updated to a higher value.

22. Additional optional basic cluster attributes and reset to Factory New command added to the existing basic cluster implementation. Updates are done in files `ciBasicCluster.h, ciClusters.h, ciIdentifyCluster.h, ciBasicCluster.c, ciClusters.c, dlBasicCluster.c, dlClusters.h, dlClusters.c, dsBasicCluster.h, dsBasicCluster.c, dsClusters.c, dsClusters .h, msBasicCluster.h, msBasicCluster.c, msClusters.c w.r.t to the new cluster attributes and command addition as detailed in the AVR2050 user guide.`

23. HA reference applications and BSP are updated with power configuration cluster addition. Updates are done in files `ciClusters.c , ciClusters.h, combinedInterface.c, dsAlarmsCluster.h, dsClusters.h, dimmerSwitch.c, dsClusters.c, dsConsole.c, bspVoltageSensor.h, bspVoltageSensor.c` with the new cluster type addition as most of the common files already mentioned in point 24 and 25. (BCZPRO-839)

24. HA reference applications are updated with diagnostics cluster addition. Changes are done in files `ciClusters.c , ciClusters.h, combinedInterface.c,ciCOnsole.c,`

clusters.h,multisensor.c,msClusters.h,msClusters.c with the new cluster type addition as most of the common files already mentioned in point 24 and 25(BCZPRO-840)

25. resetToFN console command in HA application was clearing security frame counters and this behavior corrected/updated as expected(i.e., not clearing them).Changes are done in files ciConsole.c, ciIdentifyCluster.c, zclDevice.c, dlConsole.c, dlIdentifyCluster.c, dsConsole.c, dsIdentifyCluster.c, msIdentifyCluster.c, msConsole.c by replacing the function call PDS_Delete(PDS_ALL_EXISTENT_MEMORY) to PDS_DeleteAll(false).(BCZPRO-1732)

26. The behavior of the HA application changed when the device receives a Network Leave command over the air now. The device would do a reset to factory new and changes for this are done in files zclDevice.c by introducing a new event by name nwkLeaveIndListener and modifying the ZDO_MgmtNwkUpdateNotf function to do the operation.(BCZPRO-1472)

27. The ZHA device now terminates its identification procedure after receiving IdentifyTime with value 0 .Updates are done in files ciIdentifyCluster.c , dlIdentifyCluster.c, dsIdentifyCluster.c and msIdentifyCluster.c by introducing a new function ZCL_IdentifyAttributeEventInd (BCZPRO-1492)