1. GENERAL DESCRIPTION
   1. Purpose of the Document

This document defines the ATP software requirements assigned from the onboard train control subsystem of the CASCO's intelligent train control production, including the constraints, safety requirements reliability, maintainability, interface specifications and all functional requirements with which ATP software need to comply.

The ATP software described in this document is not only including the software running on the VLE-2 board deployed in the onboard controller rack, but including the communication interface with other systems.

This document is the specification for the ATP software designers, and is the reference for safety analysts, valuators, project managers and requirement verifiers etc.

* 1. Field of Application

This document is part of iTC R&D, and it concerns the Carborne Controller, which is an onboard ATC part of the iCMTC signaling solution.

* 1. Identification of the Last Requirement Defined

The last requirement defined is the ATP-SwRS-0804, and the numeration must restart from ATP-SwRS-0805.

1. SYSTEM CONTEXT
   1. Purpose

The ATP software is the core component of the Carborne Controller, playing the important role for guaranteeing the safety of the train during operation. ATP achieved including:

using hardware board resources to realize the period time control, and the 2oo2 data comparison;

access external systems or data from other subsystems within the CC;

through from the rolling stock and internal or external systems data to determine the state of ATP, the EOA and dynamic track profiles;

according to the odometer and trackside beacon message, calculating the train's kinematics and localizing the train on the map;

monitoring train speed, location and energy to avoid the train to exceed the environmental restricted, and ensuring the train to stop before the end of authorized;

according to the platform information, enabling to operate the train door and PSD;

generating vital reports and commands to external systems and the rolling stock.

* 1. Assumptions

ATP software developers should be aware of the basic principles of urban rail transit signaling system, familiar with carborne controller's requirements and work environment, understand the hardware structure and functions of each board, experience with real-time embedded systems development, and follow the relative standards of the railway signaling production.

The vital part of ATP software applies "Vital Coded Processor " technique, which can convert the raw ATP source codes to the vital coded codes, and guarantee that the occur of random failures by the hardware must lead to the vital code error under the probability of SIL4, and the vital code errors can be detected by the checking mechanism.

The onboard ATP software running on CASCO's new 2×2oo2 vital computation platform. The hardware platform should have sufficient resources to meet the performance requirements of ATP software. Meanwhile, the platform should provide 2 channels for redundant communication and independent computation.

* 1. Application and Reference Documents
  2. Abbreviations and Defines

Reference to [APP1] for abbreviations and defines.

* 1. Conventions

The organization of ATP software development shall follow the process of SIL4 system defined in EN50128. The requirements of ATP inherit from the CC subsystem architecture description, and defining a unique label for each requirement is used to guarantee the traceability with the higher level demands. In this specification, ATP requirements are partitioned as different functional modules; and for each module, the inputs and outputs shall be defined clearly. Each functional requirement of ATP shall be testable, which means the document shall give a precise definition for each requirement. Therefore, for increasing the accuracy, ARDL (Accurate Requirement Description Language) are defined and used to describe the procedure of a requirement. In this way, tester easy to set test conditions according to the ARDL and observe whether the requirement is realized correctly.

The conventions of ARDL used for describing software requirements are shown as following:

Use indentation to express hierarchy structure;

Use (k) after each variable to represent the value in the current cycle, and (k-1) for the last cycle;

Use “and”, “or”, “not” for logic operation;

Use brackets to represent arrays;

Use uppercase to represent system constants.

The description of common functions used in ARDL are shown as [Table 23](#Ref3108492881).

1. GENERAL CONSTRAINTS
   1. User Characteristics

N/A

* 1. Safety Requirements

[iTC\_CC\_ATP-SwRS-0001]

Refer to [6.7](#bookmark30), the onboard ATP software shall be SIL4 according to the CENELEC standard [[REF16]](#bookmark31) and [[REF17]](#bookmark32).

* 1. Production Constraints

[iTC\_CC\_ATP-SwRS-0003]

ATP software shall use parameters come from iTC system data preparation for different project's deployment. For each project, ATP software reads the track map and vital setting to adapt the features of train and system design.

[iTC\_CC\_ATP-SwRS-0004]

For sub-system design constraints, it is not possible to establish more than two channel of communication with PSD platform manager. Therefore, ATP shall activate a maximum of two communication channels with PSD platform managers ordered by CC Non Vital.

[iTC\_CC\_ATP-SwRS-0005]

For interface design constraints, ATP shall use FSFB2 safety protocol with PSD platform managers with respect with safety principles.

[iTC\_CC\_ATP-SwRS-0550]

During the operation, the onboard ATP shall not response any request or command from the VLE-2 debugging Ethernet port, in order to avoid the invalid accesses and the attacks.

[iTC\_CC\_ATP-SwRS-0498]

The codes of the onboard ATP shall follow the [[REF15]](#bookmark36) rules of the VCP.

NOTES:

For each vital variable encoded by the VCP tool, its higher 32 bits are raw data, and its lower 32 bits are check words of the higher data. The check words involve time-label technique.

[iTC\_CC\_ATP-SwRS-0499]

For VCP constraints, the onboard ATP importing data from non-vital devices shall encode firstly. The data of project configuration, track map and CC data plug, all coded by the VCP offline tool.

[iTC\_CC\_ATP-SwRS-0500]

Each set of onboard ATP shall have two parts of programs, which compose two out of two structures. One program deploys on one of two CPUs of the VLE-2 board, and the other deploy on another CPU. They perform similar functions but encode by iCODER100-VCP1 and iCODER100-VCP2 respectively.

[iTC\_CC\_ATP-SwRS-0501]

The onboard ATP shall work on the VxWorks 5.5 operating system.

[iTC\_CC\_ATP-SwRS-0760]

iCODER-VCP1 and iCODER-VCP2 shall be installed in two different computer. The two redundant codes generated by independent VCP1 and VCP2 must be fit through comparison before being used. Two different commercial compare tools have been used to compare the redundant code generated by two channel.

[iTC\_CC\_ATP-SwRS-0761]

The number of global variables that need to be redundant coded in the ATP source code cannot be more than 3000 and the function number is less than 300 and the local variables less than 20 (the structure local variables can be calculated by the member numbers of the struct) and the Boolean variables less than 100.

[iTC\_CC\_ATP-SwRS-0762]

The redundant code generation function can generate redundant code for functions which satisfies the following constrains of calling order: the module function (Not include the module function which noted by comment key word /\*KWNOPARMF\*/, this kind of module function is processed as compound statement.) can only be called at the top level of the program, and the coded order of the module function must be the same with the execution order in source code. If the order of module function call at running phase is fun1 first and then fun2, then the redundant coding order shall also first code to fun1, then encode fun2.

[iTC\_CC\_ATP-SwRS-0763]

The following four annotations of key words are supported:

The function definition which noted by comment key word /\*KWNOPARMF\*/, means that the function call will be replaced by the function body. And use restrictions for comment key word /\*KWNOPARMF\*/ are as following:

The function must be defined before it is firstly be called, otherwise, give an error message when redundant code signature checking;

The function cannot have parameters. Otherwise, give an error message.

Return statement cannot be used in this function. Otherwise, give an error message.

Local variables cannot be used in this function. Otherwise, give an error message.

The use restrictions for comment keyword /\*KWADJMF\*/ are as following.

The module function which be noted must satisfy the restrictions of module function. Otherwise, give an error message.

The module function which be noted by keyword /\*KWADJMF\*/ must be coded separately. Otherwise, give an error message when redundant code signature checking.

/\*KWADJMF\*/ can only be used to note module function call. And when coding this module function, the global variables of the module function must use the pre-assigned signatures at the entrance of the function. And after every operation and adjustment, the signature of the variable must adjust to the signature which in the pre-assigned signature table. If the function is not be configured, give an error message.

The total number of signatures which used in the module function noted by keyword /\*KWADJMF\*/ cannot exceed 5000. Otherwise, give an error message.

This module function cannot use array. Otherwise, give an error message.

This module function can only use once in the coding files. Otherwise, give an error message.

These fields annotated by keyword are only valid when we first encounter the keyword name, and become invalid when encounter the second one, and unrelated with different nesting relationship between the independent keyword. But the comment type keyword /\*KWCODEDBOOL\*/ can not nest used with /\*KWNOADJ\*/, /\*KWNOPARSE\*/ or /\*KWNODT\*/. And the comment keyword /\*KWNOPARSE\*/ cannot nest with /\*KWNOADJ\*/, /\*KWCODEDBOOL\*/ or /\*KWNODT\*/. Otherwise the iCODER-100 will make an error report.

[iTC\_CC\_ATP-SwRS-0764]

For the system which using iCODER-100 to achieve SIL requirement, the signature of the output shall be critically checked.

[iTC\_CC\_ATP-SwRS-0765]

Header file’s nesting cannot exceed 10 layers.

[iTC\_CC\_ATP-SwRS-0766]

The minimal configuration of the hardware that the software needs:

* 1. Quality Requirements

N/A

* 1. Performance

[iTC\_CC\_ATP-SwRS-0008]

Vital inputs of ATP embedded software shall be acquired every ATP\_CYCLE\_TIME.

Once vital inputs acquired, ATP embedded software shall refresh its outputs in less than two ATP\_CYCLE\_TIME, and send them to CCNV.

[iTC\_CC\_ATP-SwRS-0575]

The set-up of the ATP software shall not exceed C\_TTIS minutes.

[iTC\_CC\_ATP-SwRS-0009]

From power-up and if conditions on inputs allow to reach a higher level of train availability (that is to compute at least one safety-related output at permissive state) ATP embedded software shall be able to increase train operation availability in less than INIT\_AVAIL\_MAX\_TIME.

1. INTERFACE SPECIFICATIONS
   1. List of Interface
      1. External interface

According to [REF2], the onboard ATP shall have external interfaces with other system, as shown in the .

* + 1. Internal interface

According to the VLE-2 structure and 2oo2 constraints, the onboard ATP shall have two internal interfaces, as shown in the [Table 42](#Ref2681570981).

* 1. Interface with CC Data Plug
     1. Role of interface

ATP through this interface to obtain the device number of the CC, train type and cab id, used to determine train characteristics as well as the network identification information.

[iTC\_CC\_ATP-SwRS-0551]

The interface between ATP and CC data plug shall be compliant with [REF3] document.

* + 1. Physical level

Refer to [REF3].

* + 1. Protocol level

Refer to [REF3].

* + 1. Application level
       1. ATP->CC data plug
       2. CC data plug->ATP

ATP read [CCdataPlugInfo](#CCdataPlugInfo) from the CC data plug whose structure as shown in Table 44. The safety-related information used by ATP are coded by VCP and the other information such IP address sent to DLU or CCNV are not need to encode.

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with VPB
     1. Role of interface

Through this interface, ATP obtains the real-time odometer and beacon information updated by the VPB board. In addition, when the train stopped, ATP sends the odometer sensors testing sequence to the VPB board.

[iTC\_CC\_ATP-SwRS-0552]

The interface between ATP and VPB shall be compliant with [REF4] document.

* + 1. Physical level

Refer to [REF4].

* + 1. Protocol level

Refer to [REF4].

* + 1. Application level

VPB board receives and processes the signal from odometer and beacon antenna, and ATP gets the result through the VPB registers. There are two independent channels in the VPB board, and it will correspond to each CPU in ATP respectively to form the structure 2 out of 2.

According to [REF4], the onboard ATP shall send information to VPB as shown in the [Table 45](#Ref2777736081).

NOTES：

For the CPU1 for the VLE-2 board, ATP software writes the sensors testing sequence to the VPB register; And for the CPU2, ATP software read the sensors testing from the VPB register.

According to the [REF4], the onboard ATP read information from VPB as shown in [Table 46](#Ref2777735581).

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with CC Non Vital
     1. Role of interface

ATP and CCNV, which run on the DVCOM-2 board, accomplish the core computation functions of the Carborne Controller. Through this interface, ATP send vital messages to the redundant ATP, VIOM as well as external systems such as the [ZC](#bookmark37), LC, CI and also received messages from these devices. In addition, ATP needs to send its running status to the CCNV for maintenance and receive CCNV's operational commands to achieve certain functions.

[iTC\_CC\_ATP-SwRS-0553]

The interface between ATP and CCNV shall be compliant with [[REF5]](#bookmark39) document.

* + 1. Physical level

Refer to [[REF5]](#bookmark40).

* + 1. Protocol level

Refer to [[REF5]](#bookmark41).

* + 1. Application level
       1. ATP->CCNV: initialization report

After power up, ATP software needs to send train id, train type and network address to the CCNV, which read from the CC data plug according to the [[REF5]](#bookmark43), and the RTC clock information read from VLE-2 board. As shown in [Table 47](#Ref2882304151).

* + - 1. CCNV->ATP: Non vital request

For some part of functions that does not regard as SIL4, CCNV shall handle them and transmit the request to ATP. These variables will not affect the safety of ATP, so it can be used as input or judging criteria for ATP calculation.

[iTC\_CC\_ATP-SwRS-0473]

Every cycle, ATP software queries whether there are non-vital message transmitted from CCNV, and ATP shall receive and store them into [NonVitalRequest](#NonVitalRequest):

If there is no non-vital message from CCNV, then do nothing;

Otherwise, ATP shall check the CRC according to [[REF5]](#bookmark45):

If the CRC is correct, then ATP generate the structure of global variable such as [Table 48](#Ref2777771851);

Otherwise, ATP shall discard this message.

The sequence of the four VIOMs in transportation shall be, in turn, VIOM1-1 and VIOM1-2 in CAB1, and VIOM2-1 and VIOM2-2 in CAB2.

* + - 1. ATP->CCNV: Non vital report

ATP needs to send the information to CCNV including cycle time, internal status, location and vital output order etc. Based on this information, CCNV executes the control functions or forward this information to DMI or other devices.

[iTC\_CC\_ATP-SwRS-0447]

Every cycle, ATP shall send the non-vital outputs[NonVitalReport](#NonVitalReport) to CCNV, whose structure is shown in [Table 49](#Ref2869087102). The rules are as follows:

At initialization, set all these information at restrictive state;

In normal operation, set the ATP status in accordance with the [Table 49](#Ref2869087103)

[iTC\_CC\_ATP-SwRS-0481]

Every cycle, ATP shall send [NonVitalReport](#NonVitalReport) message to CCNV

* + - 1. ATP->CCNV maintenance information

[iTC\_CC\_ATP-SwRS-0742]

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with Distant ATP
     1. Role of interface

[iTC\_CC\_ATP-SwRS-0602]

The interface between ATP and distant ATP shall be compliant with [[REF6]](#bookmark48) document.

* + 1. Physical level
    2. Protocol level
    3. Application level
       1. Local ATP->Distant ATP

[iTC\_CC\_ATP-SwRS-0372]

At each cycle, ATP shall send a [CCsynchroReport](#CCsynchroReport), which structure is [ST\_SYNCHRO\_REPORT](#bookmark52), to the redundant ATP located at the other cab.

According to [[REF6]](#bookmark55), ATP sends the synchronization message to the redundant ATP. The structure of [ST\_SYNCHRO\_REPORT](#bookmark56) shows in [Table 410](#Ref2777764342).

[iTC\_CC\_ATP-SwRS-0475]

Every cycle, ATP software sends [IdenticalCCsyncReport](#IdenticalCCsyncReport) message to CCNV according to [[REF6]](#bookmark58) as the SACEM communication protocol. The CCNV will forward this message to the redundant ATP.

* + - 1. Distant ATP -> Local ATP

According to [[REF6]](#bookmark61), ATP software should receive the synchronization report from the redundant ATP, as shown in [Table 410](#Ref2777764345).

[iTC\_CC\_ATP-SwRS-0469]

Every cycle, ATP software queries whether there is redundant ATP message [CCNV\_RedundantATPmessage](#CCNV_RedundantATPmessage) transmitted from CCNV. ATP shall receive and store the message into [OtherCCsynchroReport](#OtherCCsynchroReport). The receiving and storing rules are following:

If there is no message from the redundant ATP, or the CRC or vital checksum of this message checked failure, ATP shall discard this message;

Otherwise, ATP shall generate the structure of global variable such as [ST\_SYNCHRO\_REPORT](#bookmark63).

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with VIOM
     1. Role of interface

[iTC\_CC\_ATP-SwRS-0603]

The interface between ATP and VIOM shall be compliant with [[REF5]](#bookmark66) document.

* + 1. Physical level
    2. Protocol level
    3. Application level
       1. VIOM->ATP

According to [[REF5]](#bookmark70), the ATP software received from both ends of the train for four VIOM vital inputs message. The vital inputs of the cab, are the result of “logical OR” operation between the two VIOM’s of the same cab. For each VIOM there are as shown in [Table 411](#Ref2777771761).

* + - 1. ATP->VIOM:

According to [[REF5]](#bookmark72), ATP send vital outputs to the four VIOM respectively and the message structure as shown in [Table 412](#Ref2777764301).

[iTC\_CC\_ATP-SwRS-0474]

Every cycle, ATP Software sends the rolling stock commands [IdenticalVIOM1Out](#IdenticalVIOM1Out) and [IdenticalVIOM2Out](#IdenticalVIOM2Out) to CCNV according to [[REF5]](#bookmark74) as the SACEM communication protocol. The CCNV will forward these messages to 4 VIOMs of both train ends

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with ZC
     1. Role of interface

[iTC\_CC\_ATP-SwRS-0604]

The interface between ATP and ZC shall be compliant with [[REF7]](#bookmark77) document.

* + 1. Physical level
    2. Protocol level
    3. Application level
       1. ATP->ZC: Location report

[iTC\_CC\_ATP-SwRS-0466]

Every cycle, ATP shall generate the Location Report message to the [ZC](#bookmark81), as shown in [Table 413](#Ref2777764431).

[iTC\_CC\_ATP-SwRS-0476]

If the [SendLocReportOnZCunderTrainHead](#SendLocReportOnZCunderTrainHead) in this cycle is True, ATP shall sends [IdenticalLocReport](#IdenticalLocReport) message to CCNV according to [[REF7]](#bookmark84) as the SACEM communication protocol. The CCNV will forward this message to the [ZC](#bookmark85) which id is [ZCidUnderTrainHead](#ZCidUnderTrainHead).

[iTC\_CC\_ATP-SwRS-0477]

If the [SendLocReportOnZCunderTrainTail](#SendLocReportOnZCunderTrainTail)this cycle is True, ATP shall sends [IdenticalLocReport](#IdenticalLocReport) message to CCNV according to [[REF7]](#bookmark87) as the SACEM communication protocol. The CCNV will forward this message to the [ZC](#bookmark88) which id is [ZCidUnderTrainTail](#ZCidUnderTrainTail).

* + - 1. ZC->ATP: End of authorization

According to the Location Reports of the CC, the [ZC](#bookmark89) calculates the farthest authorization location of the train, as the EOA of this CC. ATP shall receive and determine the validity of the EOA. In CBTC mode, if no available EOA from the [ZC](#bookmark90), the ATP shall prevent train moving.

According to [[REF7]](#bookmark93), ATP software should receive the EOA message from the [ZC](#bookmark94), as shown in [Table 414](#Ref2777772391).

[iTC\_CC\_ATP-SwRS-0470]

Every cycle, ATP software queries whether there are [ZC](#bookmark98) messages [CCNV\_EOAmessage](#CCNV_EOAmessage) and [CCNV\_VariantMessage](#CCNV_VariantMessage) transmitted from CCNV. ATP shall receive and store the message into [EOA\_Report](#EOA_Report) and [VariantReport](#VariantReport), and get the SSID from these messages to generate *SSIDofZC*. The receiving and storing rules are following:

If there is no message from [ZC](#bookmark99), or the CRC or vital checksum of this message checked failure, ATP shall discard this message;

Otherwise, ATP shall generate the structure of global variable such as [ST\_EOA\_REPORT](#bookmark100) and [ST\_VARIANT\_REPORT](#bookmark101).

NOTES:

Considering the time consumption for the checksums calculating of variant reports, the rule that one [ZC](#bookmark102) can manage at most two line sections shall be followed during the project data preparation. Therefore, in one variant report message, there are maximum four line sections' variants (two for current [ZC](#bookmark103), and each one for the downstream and the upstream adjacent [ZC](#bookmark104)) sent from the [ZC](#bookmark105).

* + - 1. [ZC](#bookmark106)->ATP: Variants

When [ZC](#bookmark107) sent EOA, includes the variants in this [ZC](#bookmark108) area and adjacent [ZC](#bookmark109) to CC. Therefore, when received a valid EOA message, ATP shall parse these variants information at the same time.

According to [[REF7]](#bookmark112), ATP software should receive the variants message from the [ZC](#bookmark113), which may include several line sections' variants. The structure of one line section's variants shows in [Table 415](#Ref2864039461).

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with LC
     1. Role of interface

[iTC\_CC\_ATP-SwRS-0605]

The interface between ATP and LC shall be compliant with [[REF8]](#bookmark117) document.

* + 1. Physical level
    2. Protocol level
    3. Application level

[iTC\_CC\_ATP-SwRS-0471]

Every cycle, ATP software queries whether there are LC messages [CCNV\_SynchrodateMessage](#CCNV_SynchrodateMessage), [CCNV\_VersionAuthMessage](#CCNV_VersionAuthMessage) and [CCNV\_TSRmessage](#CCNV_TSRmessage) transmitted from CCNV. ATP shall receive and store these messages into [DateSynchronizationReport](#DateSynchronizationReport), [VersionAuthorization](#VersionAuthorization) and [TSRdownloadContent](#TSRdownloadContent):

If there is no message from LC or the CRC, or vital checksum of the message checked failure, ATP shall discard this message;

Otherwise, ATP shall generate the structure of global variable such as [ST\_DATE\_SYNCH\_REPORT](#bookmark124), [ST\_VERSION\_AUTH](#bookmark125) and [ST\_TSR\_BLOCK](#bookmark126).

If there is no [CCNV\_TSRmessage](#CCNV_TSRmessage) or the number of TSR in this message is more than MAX\_TSR\_NUMBER, ATP shall not generate *TSRdownloacContent*.

* + - 1. LC->ATP: Trackside date

According to the date synchronization information from the LC, the ATP maintains the trackside time, and sends it to [ZC](#bookmark127).

According to [[REF8]](#bookmark130), ATP software should receive the synchronization message from the LC, as shown in [Table 416](#Ref2777772961).

* + - 1. LC->ATP: Temporary speed restriction

In CBTC mode, ATP shall consider the temporary speed restriction setting on the track. When LC just power on, the message sent to ATP is not including the TRS parts until the LC received the TRS release command from the ATS operator. At that time, the ATP software shall consider all track map has assigned the most restrictive temporary speed restriction. Similarly, if the ATP has long time not received information from the LC, it shall consider the same restrictive TSR are applied.

According to [[REF8]](#bookmark133), ATP software should receive the TSR message from the LC, as shown in [Table 417](#Ref2864043851).

* + - 1. LC->ATP: Version authorization

The ATP shall send the versions of its software and data to the LC, who uses this information to judge whether to authorize CC running on the relevant track.

According to [[REF8]](#bookmark136), ATP software should receive the version authorization message from the LC, as shown in [Table 418](#Ref2864044091).

* + - 1. ATP->LC: Version report

According to [[REF8]](#bookmark138), ATP sends version report to the LC, as shown in .

CCNV is responsible for the communication management with LC. After adding the cycle time and checksum, ATP shall collect and send the information to the CCNV, which include the current software version, the version of the configuration data, the version of the all the [ZC](#bookmark139) region in the track map,

[iTC\_CC\_ATP-SwRS-0443]

The ATP shall provide the [VersionFromCCreport](#VersionFromCCreport) message to the CCNV, who will forward them to the LC. The version of the following vital components as :

ATP software version

ATP vital parameters version

ATP used [ZC](#bookmark140) areas versions (Up to 16 versions maximum)

ATP cycle time

[iTC\_CC\_ATP-SwRS-0480]

Every cycle, ATP software shall send [IdenticalVersionReport](#IdenticalVersionReport) to CCNV according to [[REF8]](#bookmark142). The CCNV will forward this message to the LC.

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with CI Radio

[iTC\_CC\_ATP-SwRS-0606]

The interface between ATP and CI shall be compliant with [[REF5]](#bookmark145) document.

* + 1. Physical level
    2. Protocol level

Refer to [[REF5]](#bookmark147).

CC→CBI: CBI variant request

CBI→CC: CBI variant report

CBI→CC: CC variant request

CC→CBI: CC variant report

* + 1. Application level
       1. ATP->CI: CBI variant request

[iTC\_CC\_ATP-SwRS-0607]

[iTC\_CC\_ATP-SwRS-0608]

* + - 1. CI->ATP: CBI variant report

[iTC\_CC\_ATP-SwRS-0609]

* + - 1. CI->ATP: CC variant request

[iTC\_CC\_ATP-SwRS-0610]

* + - 1. ATP->CI: CC variant report

[iTC\_CC\_ATP-SwRS-0611]

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with PSD

[iTC\_CC\_ATP-SwRS-0612]

The interface between ATP and PSD shall be compliant with [[REF5]](#bookmark153) document.

* + 1. Physical level
    2. Protocol level

CC→CBI: PSD "Setting"

CBI→CC: PSD "IO Status"

CC→CBI: SSE realignment

CC→CBI: SSR realignment

CBI→CC: SSE realignment

CBI→CC: SSR realignment

* + 1. Application level
       1. ATP->CI: PSD setting

According to [[REF5]](#bookmark156), ATP sends PSD setting message *CIsetting* to the CI, as shown in [Table 424](#Ref3441209441).

[iTC\_CC\_ATP-SwRS-0478]

According to the status of [CommunicateWithPSD](#CommunicateWithPSD), the conditions of establishment are as follow:

If the [CommunicateWithPSD](#CommunicateWithPSD) at previous cycle was **False** and become **True** in this CYCLE, ATP shall establish connection with CI as the FSFB2 communication protocol.

Or else:, if the [CommunicateWithPSD](#CommunicateWithPSD) is **True**, ATP shall maintain the connection with CI.

If the connection with CI is interrupted, ATP shall reestablish the communication.

Otherwise, if the [CommunicateWithPSD](#CommunicateWithPSD) is **False**, ATP shall end the connection.

[iTC\_CC\_ATP-SwRS-0479]

When the communication between ATP and CI has established, ATP software every [ATPsetting](#ATPsetting).PSDcommCycle cycle sends the PSD control command [IdenticalCIsetting](#IdenticalCIsetting) to CCNV according to [[REF5]](#bookmark159) as FSFB2 protocol.

IF [PSDmanagerOrder\_A](#PSDmanagerOrder_A).Id is valid, ATP sends command to [IdenticalCIsetting](#IdenticalCIsetting)[0];

Else: if [PSDmanagerOrder\_B](#PSDmanagerOrder_B).Id is valid, ATP sends command to [IdenticalCIsetting](#IdenticalCIsetting)[1];

* + - 1. CI->ATP: PSD IO status

[iTC\_CC\_ATP-SwRS-0472]

ATP software queries whether there is CI message transmitted from CCNV at every [ATPsetting](#ATPsetting).PSDcommCycle. ATP shall receive and store the message into [CI\_IOstatus](#CI_IOstatus):

If there is no message from CI, or the CRC, or vital checksum of this message checked failure:

If the communication with CI succeeded on previous cycles and the age is not more than **FSFB2\_MESSAGE\_TIMEOUT**, ATP shall still use the valid previous CI message to update [CI\_IOstatus](#CI_IOstatus);

Otherwise, ATP shall set all CI status as restricted.

Otherwise, ATP shall generate the structure of global variable such as [Table 425](#Ref2777773193).

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with DLU
     1. Role of interface

This interface is used to ATP sending network addresses to the DLU just after power-up.

[iTC\_CC\_ATP-SwRS-0554]

The interface between ATP and DLU shall be compliant with [[REF12]](#bookmark161) document.

* + 1. Physical level

Refer to [[REF12]](#bookmark162).

* + 1. Protocol level

Refer to [[REF12]](#bookmark163).

* + 1. Application level

During initialization, ATP software should send the network addressed to DLU as shown in [Table 426](#Ref2855538441).

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with Project Data
     1. Role of interface

This interface is used by the ATP software to read project vital settings and track maps, which generated by CC offline tool from project configurations.

[iTC\_CC\_ATP-SwRS-0555]

The interface between ATP and project data shall be compliant with [[REF11]](#bookmark165) document.

* + 1. Physical level

Refer to [[REF11]](#bookmark167).

* + 1. Protocol level

Refer to [[REF11]](#bookmark169).

* + 1. Application level

According to the [[REF11]](#bookmark171), the CC offline tool should read project configurations, generate the binary project vital settings and track maps, as shown in [Table 427](#Ref2869367521).

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface between two CPUs
     1. Role of interface

Two ATP software run on two PowerPC CPUs of the VLE-2 board respectively. They are used different VCP signature tables to avoid common mode failure. The two CPUs connected by dual-ram and software can use this interface to synchronize their cycle time.

* + 1. Physical level

Refer to [[REF9]](#bookmark173).

* + 1. Protocol level

Refer to [[REF9]](#bookmark175).

* + 1. Application level

[iTC\_CC\_ATP-SwRS-0025]

Through the DPRAM between two CPUS of the VLE-2 board, ATP software should send information listed in [Table 428](#Ref2777809931) to the ATP on the other CPU.

[iTC\_CC\_ATP-SwRS-0026]

Through the DPRAM between two CPUS of the VLE-2 board, ATP software should read information listed in [Table 429](#Ref2801818691) from the ATP on the other CPU.

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with Memorized Location
     1. Role of interface

Memorized location function used to improve the efficiency and safety in parking are——after the power-on, train can directly running under the ATP protection. When the train completely stopped in sleeping areas, ATP shall record the location and does not lost during power off. And after power-on again, the record can be read as the initialization position of the train.

* + 1. Physical level

Will be involved in ATP software design document.

* + 1. Protocol level

Will be involved in ATP software design document.

* + 1. Application level

The location information need to memorize are shown in [Table 430](#Ref3456762201).

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

* 1. Interface with VLE Hardware
     1. Role of interface

ATP software runs on the VLE-2 board, so it should use the hardware resources including vital clock, file system etc.

* + 1. Physical level

Refer to [[REF9]](#bookmark177).

* + 1. Protocol level

Refer to [[REF9]](#bookmark179).

* + 1. Application level

[iTC\_CC\_ATP-SwRS-0028]

ATP can obtain the hardware resources shown in [Table 431](#Ref2777831311).

* + 1. Software compatibility

N/A

* + 1. Hardware compatibility

N/A

* + 1. Implicit choices and justification

N/A

1. FUNCTIONAL REQUIREMENTS
   1. Description of Functions

According to the level of the ATP functionality, the software requirements divided into several functional blocks, as shown in [Table 51](#Ref2400779981).

* 1. Description of Data
     1. Constants for iTC production

As a part of iTC, ATP shall comply with the capacity limit of the system. Therefore, ATP software shall able to handle such a scale of information defined in Table5-2.

* + 1. Configurable data for projects

For application to different projects, ATP shall load vital settings before cycling operation, as shown as [Table 61](#Ref2831100001), the configurable data including the parameters of track, specific demands for the project, train characteristics etc. Refer to [[REF11]](#bookmark181) for the detail definition.

* + 1. Track map

The track map onboard ATP used come from project configuration. As shown in [Figure 51](#Ref3475673591), the basic constitution unit of track map is block, which managed by ZC; on the block, a series of [singularities](#bookmark183) represent physical devices such as signal, switcher, platform, as well as permanent speed restrictions, protected areas and other virtual restricted zones or limit points.

According to the design of track map, an abscissa of a block means the distance from the block endpoint of the **UP** orientation, as shown in [Figure 52](#Ref3475673601). The abscissa of the block starts from zero, the **UP** orientation endpoint, and increases along the **UP** orientation until reached the length of this block, the **DOWN** orientation endpoint. That is, in a same block, towards **UP** orientation, the larger of the abscissa value, the upper of the location; and vice versa. If an abscissa exceeds the length of the block, the actual location should be in the downstream block on the **UP** orientation; other hand, if an abscissa is less than zero, it should be in the downstream block on the **DOWN** orientation.

The **UP** or **DOWN** orientation in track map is defined by project.

Some of the singularities, there status will change dynamically, such as switch positions, permissive or restrictive signals.

* 1. F1- Manage System Information
  2. F11-Acquire Configuration Data
     1. Inputs
     2. Locals
     3. Processing
        1. Get CC data plug information

In the same project, the ATP software running on different vehicles is identical. When communicating with external systems, the ATP depends on the CC SSID to identify itself; meanwhile, it needs to know itself in the train **END\_1** or **END\_2**; and runs in the CPU1 or CPU2 module of the VLE-2 board. These information are stored in CC data plug where installed on each CPU module of the VLE-2 board. The maintenance staff guarantees the correctness and uniqueness of CC data plug.

[iTC\_CC\_ATP-SwRS-0053]

On initialization, ATP generates [TrainType](#TrainType) according to [DataPlugContent](#DataPlugContent).CCTrainType from the CC data plug.

[iTC\_CC\_ATP-SwRS-0054]

On initialization, ATP generates [CoreId](#CoreId) according to [DataPlugContent](#DataPlugContent).[CCCoreId](#bookmark184) read from the CC data plug.

[iTC\_CC\_ATP-SwRS-0613]

Core id for CC on the distant cab.

[iTC\_CC\_ATP-SwRS-0055]

On initialization, ATP generates [SubSystemId](#SubSystemId) according to [DataPlugContent](#DataPlugContent).CC\_SSID from the CC data plug.

* + - 1. Check the project setting

ATP software should apply to different projects. For project-related configuration parameters, track maps and other information, generated by CC offline tool, are stored in the VLE-2 board designated storage space. When power on, the ATP software shall read these configurations and perform for related functions. Before using, the ATP shall verify the correctness of these parameters and track maps, and shall ensure the consistency of these data with the other CPU module and with the redundant ATP on the other END of train.

[iTC\_CC\_ATP-SwRS-0057]

According with train type, ATP software shall compute the VCP check words of the project vital setting [DATA.VES](#bookmark186), which generated by CC offline tools.

If check words are correct, ATP software shall read the vital setting and convert to the global variable [ATPsetting](#ATPsetting) according to the [Table 61](#Ref2831100003);

Otherwise, the ATP shall cease to run.

[iTC\_CC\_ATP-SwRS-0058]

According with train type, ATP software shall read [DATA.VES](#bookmark189) and detect the track map data with VCP which was generated by CC offline tools. If the data is correct, ATP shall generate [TrackMap](#TrackMap) with the structure as refer to [[REF11]](#bookmark190); Otherwise, ATP cannot continue to operate.

[iTC\_CC\_ATP-SwRS-0755]

Some parts of codes in [DATA.VES](#bookmark193), related to input and output functions, as project configuration data, ATP links them during initialization and generates *Offline* structure as [Table 63](#Ref3477492231), refer to [[REF11]](#bookmark194).

**NOTE：**

For the configuration parameters and track map, generated by the CC offline tool, if there is error for the ATP reading, or there is any changes during the processing, the VCP coding calculation will be conducted the wrong data, which will causing the VIOM output is set as restricted status.

[iTC\_CC\_ATP-SwRS-0536]

[SafeApplicationVersion](#SafeApplicationVersion) stands for the version of ATP software. The version of ATP software is defined in source code. The version shall be update whenever the code changed.

[iTC\_CC\_ATP-SwRS-0537]

After ATP gets the offline parameters and track map from the CC data plug, it needs to compare this information with the other CPU module, which represented by the term of [IdenticalVersionOfDualCPU](#IdenticalVersionOfDualCPU). If the information is as same, ATP shall set [IdenticalVersionOfDualCPU](#IdenticalVersionOfDualCPU) as **True**; otherwise, it will set [IdenticalVersionOfDualCPU](#IdenticalVersionOfDualCPU) as **False.**

[TrainType](#TrainType), the train type from data plug;

[CoreId](#CoreId), the train end identification from data plug;

[SubSystemId](#SubSystemId), the train identification from data plug;

[TrackMap](#TrackMap), the version number of each [ZC](#bookmark195) region in line map;

[ATPsetting](#ATPsetting), the version number for the configuration data;

[SafeApplicationVersion](#SafeApplicationVersion), the version of ATP software.

[iTC\_CC\_ATP-SwRS-0059]

After all above-mentioned information has corrected, and correspondingly, ATP will generate [TrainKnown](#TrainKnown) information. If [TrainKnown](#TrainKnown) considered as **False**, ATP shall set all output ports as restricted.

* + - 1. Parse the CCNV information

[iTC\_CC\_ATP-SwRS-0125]

Through the communication with CCNV, ATP judges [NonVitalRequest](#NonVitalRequest)received from CCNV and generates [NonVitalRequestReady](#NonVitalRequestReady) If received a new message.

[iTC\_CC\_ATP-SwRS-0126]

[ATOcontrolTimeValid](#ATOcontrolTimeValid) stands for the effectiveness of CCNV message. If there is no updating CCNV message past the **CCNV\_VALIDITY\_CYCLES**, [ATOcontrolTimeValid](#ATOcontrolTimeValid) is set as **False**.

* + 1. Outputs
  1. F12-Manage Train Status
     1. Inputs
     2. Locals
     3. Processing

Every cycle, ATP software queries whether there are vital rolling stock inputs messages transmitted from CCNV, which acquired by at most four VIOM from both **END\_1** and **END\_2**. The structure of the inputs message are [Table 411](#Ref2777771764), and ATP shall receive and store them into *VIOM1VitallInput*.or *VIOM2VitalInput*. The receiving and storing rules are following:

If there is no message from VIOM, or the CRC or vital checksum of the received message failed, ATP shall discard this message;

Otherwise, generate the structure of global variable such as [ST\_VIOM\_INPUT](#bookmark202).

* + - 1. VIOM1 -> ATP

[iTC\_CC\_ATP-SwRS-0468]

ATP determines whether received a safety input message from VIOM1.

During the calculation, the value viomId is either zero or one, which represents one of the two VIOM in the train **END\_1**.

[iTC\_CC\_ATP-SwRS-0060]

Through the communication, ATP gets the vital input message from VIOM1 and decides the time effectiveness and the correctness of the sequence of the message, which defined as [VIOM1VitalInputsAvailable](#VIOM1VitalInputsAvailable).

During the calculation, the value ViomId is either zero or one, which represents one of the two VIOM in the train **END\_1**.

[iTC\_CC\_ATP-SwRS-0740]

Records the survival time of received vital inputs from VIOM1.

[iTC\_CC\_ATP-SwRS-0572]

ATP records the latest cycle time information of VIOM in **END\_1** by the term [LatestVIOM1LoopHourVIOM](#LatestVIOM1LoopHourVIOM).

In initialization, set [LatestVIOM1LoopHourVIOM](#LatestVIOM1LoopHourVIOM) as the zero;

If receiving an available VIOM1 message, or a new message and the previous one has invalid, ATP will set the related value of [LatestVIOM1LoopHourVIOM](#LatestVIOM1LoopHourVIOM) as the viomLoopHour of the message.

Otherwise, [LatestVIOM1LoopHourVIOM](#LatestVIOM1LoopHourVIOM) keeps unchanged.

* + - 1. VIOM2 -> ATP

[iTC\_CC\_ATP-SwRS-0614]

ATP determines whether received a safety input message from VIOM2.

[iTC\_CC\_ATP-SwRS-0449]

Through the communication, ATP gets the vital input message from VIOM2 and decides the time effectiveness and the correctness of the sequence of the message, which defined as [VIOM2VitalInputsAvailable](#VIOM2VitalInputsAvailable).

During the calculation, the value i is either zero or one, which represents one of the two VIOM in the train **END\_2**.

[iTC\_CC\_ATP-SwRS-0741]

Records the survival time of received vital inputs from VIOM2.

[iTC\_CC\_ATP-SwRS-0573]

ATP records the latest cycle time information of VIOM in **END\_2**by the term [LatestVIOM2LoopHourVIOM](#LatestVIOM2LoopHourVIOM).

In initialization, set [LatestVIOM2LoopHourVIOM](#LatestVIOM2LoopHourVIOM) as the zero;

If receiving an available VIOM1 message, or a new message and the previous one has invalid, ATP will set the related value of [LatestVIOM2LoopHourVIOM](#LatestVIOM2LoopHourVIOM) as the viomLoopHour of the message.

Otherwise, [LatestVIOM2LoopHourVIOM](#LatestVIOM2LoopHourVIOM) keeps unchanged.

* + - 1. Identical VIOM inputs

ATP shall perform logical "or" operation between two VIOM messages at each cab, and generate one identical VIOM input message for each cab, as shown in [Table 54](#Ref3475678511). The message as an interface, are available to offline generated code, defined in [[REF11]](#bookmark210).

[iTC\_CC\_ATP-SwRS-0538]

ATP determines whether the vital inputs message from VIOM1 valid.

[iTC\_CC\_ATP-SwRS-0539]

ATP determines whether the vital inputs message from VIOM2 valid.

[iTC\_CC\_ATP-SwRS-0061]

For four vital input messages from VIOM1 and VIOM2, ATP shall deal with it comprehensively and generate the [IdenticalInputs](#IdenticalInputs) information in this cycle for calculation of other modules, structured as . For both of VIOM message in the same END, ATP shall consider the reliability and use “logic OR” for judge the status of each port:

If both of VIOM message from one train end is invalid, the [IdenticalInputs](#IdenticalInputs) of this end is also ineffective.

If only one VIOM message from one train end is invalid, the [IdenticalInputs](#IdenticalInputs) is equal to this effective VIOM message.

If there is at least one VIOM message of a port is on permissive, ATP shall regard the corresponding port of this train END as permissive status.

Otherwise, ATP shall regard this port as restrictive status in the [IdenticalInputs](#IdenticalInputs).

According to the project configuration, the value from different input ports can participate in logic operation to get desired results, as shown in [Figure 54](#Ref3450641181).

All configurable inputs ATP supported are shown in [Table 55](#Ref3424614761). All these variables are generated by offline tools and as a part of the project data read by ATP during initialization.

* + - 1. Train coupled status management

[iTC\_CC\_ATP-SwRS-0075]

[CoupledByEnd1](#CoupledByEnd1) or [CoupledByEnd2](#CoupledByEnd2) shows that both ends of train connect with other trains. If the project is not configured with the capture of coupling input, it is certain that the train does not connect with other trains.

[iTC\_CC\_ATP-SwRS-0802]

TrainNotCoupled

[iTC\_CC\_ATP-SwRS-0789]

TrainCoupledType

**TRAIN\_COUPLED\_UNKNOWN**

**TRAIN\_NO\_COUPLED**

**TRAIN\_COUPLED\_END1**

**TRAIN\_COUPLED\_END2**

[iTC\_CC\_ATP-SwRS-0790]

CoupledTypeInconsistent

[iTC\_CC\_ATP-SwRS-0792]

MatchRebootCondition

* + 1. Outputs
  1. F13-Manage Loop Hour with Distant ATP

This module handles information from the redundant ATP, and generates loop hour as the basis of validity judgment.

* + 1. Inputs
    2. Locals
       1. ProcessingLoop hour management

Since power up, ATP shall maintain its loop hour as a label used to monitor the timeline of communication, according to the cycle number and the cab where ATP settled. For both ends of the cab of the ATP, they use different loop hour initial value, and there is no intersection between the ranges. Thus, according to the message loop hour, the source of the message sent from which ATP can distinguish.

[iTC\_CC\_ATP-SwRS-0144]

[ATPtime](#ATPtime) stands for the ATP loop hour of this train END.

Based on [CoreId](#CoreId), ATP initialize [ATPtime](#ATPtime) as the initiative value of **END\_1** or **END\_2**;

If the value exceeds the maximum loop hour, ATP shall set it as the initiative value;

Otherwise, add one for each cycle.

* + - 1. loop hour management for the other ATP

The ATP software needs to maintain the redundant ATP cycle time for monitoring the effectiveness of the message sent to the redundant side. Taking into account the transmission delay time, the redundant ATP time should be within the maximum and minimum range.

[iTC\_CC\_ATP-SwRS-0615]

OtherATPmessageReceived

The message transmitted from the distant ATP in the other END shall be protected by check words. And before using the information, ATP shall verify the check words.

[iTC\_CC\_ATP-SwRS-0078]

OtherATPmessageAvailable

[OtherATPmessageAvailable](#OtherATPmessageAvailable), ATP shall judge the effectiveness of message from the redundant ATP, shown as following pseudo-codes:

[iTC\_CC\_ATP-SwRS-0616]

LastOtherATPmessageAge

[iTC\_CC\_ATP-SwRS-0540]

[OtherATPmessageValid](#OtherATPmessageValid) represents the effectiveness of the messages from redundant ATP. If this message is invalid, ATP will set [OtherATPmessageValid](#OtherATPmessageValid) as **False**; otherwise, it is set as **True**.

[iTC\_CC\_ATP-SwRS-0081]

The [OtherATPminTime](#OtherATPminTime) stands for the local ATP maintained minimum time of the redundant ATP. The setting rule is as following:

In initialization, ATP set the [OtherATPminTime](#OtherATPminTime) as default value based on the [CoreId](#CoreId) of the redundant ATP.

Or else:, if the updating message from the new redundant ATP in this cycle is available, ATP will update [OtherATPminTime](#OtherATPminTime) as the current time in the message.

Or else:, if the redundant ATP message is still effective, ATP will add 1 in the [OtherATPminTime](#OtherATPminTime) until it is out of bound, and set is as initialization value.

Or else:, If the received a new redundant ATP message, but it was not available, ATP shall update [OtherATPminTime](#OtherATPminTime) as in the message.

Otherwise, accumulate [OtherATPminTime](#OtherATPminTime).

[iTC\_CC\_ATP-SwRS-0083]

The [OtherATPmaxTime](#OtherATPmaxTime) stands for the local ATP maintained maximum time of the redundant ATP. The rules to update [OtherATPmaxTime](#OtherATPmaxTime) are similar with [OtherATPminTime](#OtherATPminTime) except that when received a new message from the redundant ATP, the [OtherATPmaxTime](#OtherATPmaxTime) shall add the maximum transmission delay in network.

* + - 1. Parse distant ATP information

[iTC\_CC\_ATP-SwRS-0080]

OtherATP, parse and store messages from the distant ATP.

In initialization or the message has expired, set all variables as default value;

when new message available, set the corresponding value from the new message;

otherwise, remain unchanged.

* + 1. Outputs
  1. F14-Manage Variants in Block Mode
     1. Inputs
     2. Locals
     3. Processing

This module mange the variants used in block mode, including the BM beacon and the CBI radio.

* + - 1. Determine block mode

[iTC\_CC\_ATP-SwRS-0067]

[BlockModeUsed](#BlockModeUsed) represents that either of train end chooses BM mode.

[iTC\_CC\_ATP-SwRS-0066]

The status of [BMvariantValidWhileTemporallyValid](#BMvariantValidWhileTemporallyValid) shows whether it is in the BM mode.

* + - 1. BM variants received from beacon

In the block mode, the variants come from the parsing of BM beacon message. For each BM beacon, has a specific orientation and can store 16 variants at most. For the content and validity of each variant is defined in the off-line data. During the block mode, ATP shall only store the last read BM variants with the same direction as the train moving.

[iTC\_CC\_ATP-SwRS-0146]

[BeaconVariantsUpdating](#BeaconVariantsUpdating) used to determine ATP whether to update the BM variants in this cycle.

If all the following conditions are fulfilled, ATP shall set [BeaconVariantsUpdating](#BeaconVariantsUpdating) as True:

The current operational mode is BLOCK MODE;

And train moved and ATP received a BM beacon in this cycle;

And the train is either not localized, or the direction of the BM variants is as same as the orientation of the train front end.

Otherwise, ATP shall set [BeaconVariantsUpdating](#BeaconVariantsUpdating) as False.

[iTC\_CC\_ATP-SwRS-0617]

BMbeaconReadAge

[iTC\_CC\_ATP-SwRS-0147]

The structure of array ATP stored [BMbeaconVariants](#BMbeaconVariants) are [ST\_BM\_VARIANT](#bookmark246). The rules to update the BM variants are as follows:

if the [BMbeaconReadAge](#BMbeaconReadAge) is larger than the [ATPsetting](#ATPsetting).VariantsBMfullValidityTime, ATP shall set all BM variants as restricted status.

Else If the [BeaconVariantsUpdating](#BeaconVariantsUpdating) is True, then ATP update the [BMbeaconVariants](#BMbeaconVariants) by the new beacon.

Otherwise, keep BMbeaconVariants unchanged.

NOTES:

There is a situation that ATP read one BM beacon as the second beacon for ATP initializing location, and the direction of this BM beacon is opposite with the train movement.

[iTC\_CC\_ATP-SwRS-0618]

BMbeaconVariantValue

[iTC\_CC\_ATP-SwRS-0148]

[UsedBMbeaconId](#UsedBMbeaconId) records the used BM variants came from which BM beacon:

When one of the following conditions fulfilled, ATP clear the [UsedBMbeaconId](#UsedBMbeaconId):

initialization,

the BLOCK MODE variant is not temporally valid,

the direction of the used BM beacon is not as same as train front orientation,

the train is not localized.

Or else:, when received a valid BM beacon, ATP update [UsedBMbeaconId](#UsedBMbeaconId);

Otherwise, keep this value unchanged.

* + - 1. BM variants received from CBI

[iTC\_CC\_ATP-SwRS-0619]

BMvariantValidLastRisingAge,

[iTC\_CC\_ATP-SwRS-0620]

CBIvariantReportReceived

[iTC\_CC\_ATP-SwRS-0621]

CBIvariantReportAvailable

[iTC\_CC\_ATP-SwRS-0622]

CBIvariantReportLastAge

[iTC\_CC\_ATP-SwRS-0623]

BMcbiVariants

[iTC\_CC\_ATP-SwRS-0624]

CBIvariantAge

[iTC\_CC\_ATP-SwRS-0625]

CBIvariantLowValidity

[iTC\_CC\_ATP-SwRS-0626]

BMcbiVariantValue

[iTC\_CC\_ATP-SwRS-0754]

AppliedCBIvariantLoopHour

* + - 1. BM variants outputs

[iTC\_CC\_ATP-SwRS-0627]

CBIvariantMoreAvailableThanBeacon

ATP shall use the more recent message from beacons and CBI radio.

[iTC\_CC\_ATP-SwRS-0628]

BMvariantValue

[iTC\_CC\_ATP-SwRS-0629]

BMvariantRemainingTime

* + 1. Outputs
  1. F15-Manage Variants in CBTC Mode
     1. Inputs
     2. Locals
     3. Processing
        1. Version authorization by LC

[iTC\_CC\_ATP-SwRS-0630]

VersionAuthorizationReceived

[iTC\_CC\_ATP-SwRS-0103]

VersionAuthorizationAvailable

[iTC\_CC\_ATP-SwRS-0453]

LastVersionReportAge

[iTC\_CC\_ATP-SwRS-0104]

ReceivedVersionMessages

[iTC\_CC\_ATP-SwRS-0631]

VersionAuthorizedByLC

* + - 1. ZC message availability

[iTC\_CC\_ATP-SwRS-0093]

SameVersionWithDistantCore

The local ATP shall compare the information from the redundant ATP to ensure the consistency, which includes versions of vital software, project configuration data and the track map.

[iTC\_CC\_ATP-SwRS-0632]

EOAReportReceived

[iTC\_CC\_ATP-SwRS-0105]

[ZCmessageReady](#ZCmessageReady) represents that an available EOA and variants message from [ZC](#bookmark276) received in this cycle.

[iTC\_CC\_ATP-SwRS-0108]

[LastEOAReportAge](#LastEOAReportAge) represents the value calculated by current ATP time minus the previous loc-report number and the EOA consuming time in [ZC](#bookmark279).

[iTC\_CC\_ATP-SwRS-0106]

[EOAgroundAge](#EOAgroundAge) stands for the number of CC cycle when receiving the EOA information.

[iTC\_CC\_ATP-SwRS-0107]

ReceivedEOAreport

* + - 1. Variants in CBTC mode

[iTC\_CC\_ATP-SwRS-0109]

[VariantGroundAge](#VariantGroundAge) shows the survival time of the variants in [ZC](#bookmark290).

[iTC\_CC\_ATP-SwRS-0633]

VariantReportReceived

[iTC\_CC\_ATP-SwRS-0110]

ReceivedVariantReport

NOTES**：**

In the CBTC mode, ATP gets the variants by parse the [ZC](#bookmark295) message. The variants sending from [ZC](#bookmark296) are categorized by line section. Moreover, it will send the line section variants in the current [ZC](#bookmark297) area and the adjacent one. ATP will record and check the variants and maintains the time effectiveness.

[iTC\_CC\_ATP-SwRS-0150]

CBTCvariantValue

ATP shall maintain the validation of CBTC variants message from ZC. if the validation timeout, ATP should set all CBTC variants to restrictive state.

[iTC\_CC\_ATP-SwRS-0634]

VariantValue

* + 1. Outputs
  1. F2-Measure Train Kinematics
  2. F21-Manage Interface with VPB
     1. Inputs
     2. Locals
     3. Processing
        1. Odometer information

The consistency of cog count and cog code makes sure the safety during the processing of coded odometer, so ATP needs to save the value of cog count and cog code in each interrupt as shown in [Table 56](#Ref3475688741), and judge it in the main task. However when the coded odometer does not move, ATP needs to driver VPB board in order to test every sensor in the odometer. In addition, ATP needs to judge the working status of odometer through the conduction status returning from the sensor.

[iTC\_CC\_ATP-SwRS-0113]

[OdometerImm](#OdometerImm) records the changes of VPBWrite register in each interrupt, structured as [Table 56](#Ref3475688744).

[iTC\_CC\_ATP-SwRS-0114]

In the interrupt, when the [TopLocValid](#bookmark300) detected by VPB board regarded as True, [OdometerImm](#OdometerImm).[TopLocValid](#bookmark301) and [BeaconMsgReady](#BeaconMsgReady) set as True. Adding [TopLocCounter](#TopLocCounter), the data from beacon savor is saved into [BeaconMsgByte](#BeaconMsgByte)[MAX\_BEACON\_DATA\_SIZE].

[iTC\_CC\_ATP-SwRS-0557]

[SensorTestFlag](#SensorTestFlag) regarded as the symbol whether ATP in CPU 1 needs to check the sensor. In the interval, if the value of [CBKWrite](#bookmark303).CogCounterReg is not changed in the continuous SENSOR\_TEST\_START\_TIME, and the [WheelFilteredStopped](#WheelFilteredStopped) was False at last cycle, the ATP of CPU1 in VLE-2 board need to set [SensorTestFlag](#SensorTestFlag) as True and send the data to the ATP of CPU2.

[iTC\_CC\_ATP-SwRS-0558]

For the ATP of CPU1 in VLE-2 board, if it sets the [SensorTestFlag](#SensorTestFlag) of pervious end of cycle as True, it need to set the value of [OdometerImm](#OdometerImm).D1/D2/D3 based on the settled sequence in each interrupt, and set the *SensorTesting* as True.

D1/2/3 shall set to POWER\_ON at every other interrupts;

In other interrupt, D1/2/3 shall be set to pseudo random value.

[iTC\_CC\_ATP-SwRS-0559]

For the ATP of CPU2, if the [SensorTestFlag](#SensorTestFlag) from CPU1 is True, it needs to read the D1/2/3 registers of VPB-2 board and obtain the current testing value of D1/2/3.

[iTC\_CC\_ATP-SwRS-0115]

During the sensor test performing, both ATP in the two CPUs of VLE-2 shall detect the consistency of the test result according to the state of C1/2/3 in the two continuous interrupt. Accordingly, ATP determines the state of three sensors:[A1](#bookmark307), [A2](#bookmark308), [A3](#bookmark309).

Based on the above three sensors’ status A1/2/3, ATP determines the status of the odometer as following conditions:

If any one of A1/2/3 is SENSOR\_WRONG, then ATP consider the odometer as error in this interrupt and set the [OdometerImm](#OdometerImm)*.*[TestResult](#bookmark316)as TEST\_INCONSISTENT;

If all of A1/2/3 are SENSOR\_CONDUCT, then ATP consider the odometer as error in this interrupt and set the [OdometerImm](#OdometerImm)*.*[TestResult](#bookmark317)as TEST\_INCONSISTENT;

If all of A1/2/3 are SENSOR\_BLOCKED, then ATP consider the odometer as error in this interrupt and set the [OdometerImm](#OdometerImm)*.*[TestResult](#bookmark318)as TEST\_INCONSISTENT;

If all of A1/2/3 are as same as the result at last interrupt respectively, then ATP consider the odometer as stop in this interrupt and set the [OdometerImm](#OdometerImm)*.*[TestResult](#bookmark319)as TEST\_STOPPING;

If the TEST\_STOPPING has lasted more than SENSOR\_TEST\_IMMOBILE\_THRESHOLD, the ATP consider the odometer standstill, and set [OdometerImm](#OdometerImm)*.*[TestResult](#bookmark320)as TEST\_IMMOBILE;

If any one of A1/2/3 is different with the result at last interrupt, then ATP consider the odometer rolling, and set the [OdometerImm](#OdometerImm)*.*[TestResult](#bookmark321)as TEST\_FLOATING.

When the odometer test result is either TEST\_INCONSISTENT or TEST\_FLOATING, ATP shall stop the sensor test and set [OdometerImm](#OdometerImm)*.*[SensorTesting](#bookmark323) as False for this interrupt, and set [SensorTestFlag](#SensorTestFlag) as False for this cycle.

NOTES**：**

Because of the design restriction, only the FPGA1 in the VPB-2 board can send the testing sequence D1/2/3 to the sensor, and meanwhile the FPGA1 will forward the info to the FPGA2 in the VPB-2 so that the CPU2 in the VLE-2 can read. Both of the FPGA in the VPB-2 board will have the same process for the sensor testing result C1/2/3.

[iTC\_CC\_ATP-SwRS-0116]

In the specific T\_LOCK\_ODOMETER, the [OdometerImm](#OdometerImm) of all intervals need to be saved into [LockedOdometer](#LockedOdometer)[ATP\_INTERRUPT\_NB] with the index as [ImmediateNb](#ImmediateNb); The value of [BeaconMsgReady](#BeaconMsgReady), [TopLocCounter](#TopLocCounter) and [BeaconMsgByte](#BeaconMsgByte) should be recorded by using [LockedBeaconMsgReady](#LockedBeaconMsgReady), [LockedTopLocCounter](#LockedTopLocCounter) and [LockedBeaconMsgByte](#LockedBeaconMsgByte)for the main task.

[iTC\_CC\_ATP-SwRS-0590]

The ATP software in different CPUs shall synchronize the information get from the VPB board, with following rules:

For [CogCounter](#bookmark330):

If the [CogCounter](#bookmark331) read by two CPUs are different at the same interrupt, ATP shall take the large one as the result;

ATP shall use the CPU's CogCode as same as [CogCounter](#bookmark332).

For [TopLocValid](#bookmark333):

Only the cog counter latched by TOPLOC is between the before and after cog counter , TOPLOC is considered valid

If the top-loc happened in adjacent interrupt between two CPUs, the former one shall use to calculate the [CogPositionBeforeTopLoc](#CogPositionBeforeTopLoc), and the latter to calculate the [CogPositionAfterTopLoc](#CogPositionAfterTopLoc);

If the top-loc difference are more than one interrupt, ATP shall consider it as invalid.

* + - 1. Beacon message

When the Beacon antenna passes the upside of beacon, it will generate the top-loc signal and send it to VPB board. While the ATP gets this signal, it will lock the odometer cog counter at this moment, and will obtain, parse and check the beacon message.

[iTC\_CC\_ATP-SwRS-0120]

[BeaconChecksumFailure](#BeaconChecksumFailure) judges whether the checksum of beacon message is correct or not.

In the main task, if [LockedBeaconMsgReady](#LockedBeaconMsgReady) is True, [LockedBeaconMsgByte](#LockedBeaconMsgByte)need to be detected, including validity of [ATCkey](#bookmark334) and calculation of the SACEM checksum of beacon.

ATP shall reject each beacon message which vital checksum is corrupted.

[iTC\_CC\_ATP-SwRS-0117]

In the main task of ATP, if the status of [LockedBeaconMsgReady](#LockedBeaconMsgReady) in the interrupt is True, and the [BeaconChecksumFailure](#BeaconChecksumFailure) is False, ATP shall set the [BeaconMessageReceive](#BeaconMessageReceive) as True; and vice versa. If ATP receives more than one beacons in this cycle, it will deal with the last beacon and based on this data to update the [BeaconMessageReceive](#BeaconMessageReceive).

[iTC\_CC\_ATP-SwRS-0118]

[BeaconCount](#BeaconCount) represents the accumulated number of received Top-loc signal from power on to current cycle.

[iTC\_CC\_ATP-SwRS-0119]

If the [BeaconMessageReceive](#BeaconMessageReceive) is True, ATP shall parse the value of [LockedBeaconMsgByte](#LockedBeaconMsgByte) and generate *BeaconMessage* with structure as [ST\_BEACON\_MSG](#bookmark336):

[iTC\_CC\_ATP-SwRS-0121]

If the status of [BeaconMessageReceive](#BeaconMessageReceive) is True, the [BeaconMessage](#BeaconMessage)*.ID* is obtained by [LockedBeaconMsgByte](#LockedBeaconMsgByte); Otherwise, keep it unchanged.

BEACON\_ID\_BITS represents the index of beacon ID in the beacon message defined in [[REF4]](#bookmark338).

[iTC\_CC\_ATP-SwRS-0122]

If the [BeaconMessageReceive](#BeaconMessageReceive) is True, the variants is come from [LockedBeaconMsgByte](#LockedBeaconMsgByte)and ATP set as [BeaconMessage](#BeaconMessage)*.Variants*[MAX\_BM\_VARIANT\_NB]; if there is no beacon read at the end of cycle, there is no changes.

BM\_VARIANTS\_BIT\_0...BM\_VARIANTS\_BIT\_15 represents the index of BM beacon variants defined in [[REF4]](#bookmark340).

[iTC\_CC\_ATP-SwRS-0123]

If the [BeaconMessageReceive](#BeaconMessageReceive) is True, the default message is judged by [LockedBeaconMsgByte](#LockedBeaconMsgByte)and ATP set the [BeaconMessage](#BeaconMessage)*.*[DefaultMessage](#bookmark342); if there is no new beacon read, it keeps unchanged.

DEFAULT\_MESSAGE\_BIT represents the index of beacon that judges default message, which defined in the [[REF4]](#bookmark348).

[iTC\_CC\_ATP-SwRS-0124]

If the [BeaconMessageReceive](#BeaconMessageReceive) is True, it is feasible to judge whether the variants are available through [LockedBeaconMsgByte](#LockedBeaconMsgByte) and ATP set as [BeaconMessage](#BeaconMessage)*.*[BlockModeVariantAvailable](#bookmark350); If there is no new beacon read, it keeps invariable.

BLOCK\_MODE\_VARIANT\_AVAILABLE\_BIT stands for the index of the beacon variants in the beacon message defined in [[REF4]](#bookmark356).

[iTC\_CC\_ATP-SwRS-0049]

At each cycle, ATP shall provide to beacon device the [ATCkey](#bookmark357) in order to be able to control message freshness.

[iTC\_CC\_ATP-SwRS-0767]

* + 1. Outputs
  1. F22-Monitor the Odometer
     1. Inputs
     2. Locals
     3. Processing

In every cycle of main task, according to coded odometer information of each interrupt latch in the last cycle, ATP software shall calculate the kinematic parameters of the wheel. That is to say, ATP shall judge the cog numbers in this cycle according to the value in the register. If ATP did not detect the move of odometer, it will check whether the wheel is completely static based on the sensor testing. According to the design of coded odometer, it is impossible that three sensors are in conduct or blocked at the same time.

[iTC\_CC\_ATP-SwRS-0592]

The ATP software needs to calculate the expected cog code with 8 bits, according to the encoding characteristic of the disc code and the direction of odometer rotation.

When the odometer initialization, the expected [CompCogCode](#CompCogCode) shall be set as initial CogCode;

Since then, for one cog rotated in interrupt, the corresponded bit shall be shift as following rules:

if the cog increased in adjacent interrupts, the [CompCogCode](#CompCogCode) shall be shift a bit toward right from high to low; shift out the lowest one and set the new highest bit as *C4array*[*C4ArrayIndex*], and update *C4ArrayIndex*.

otherwise, if the cog decreased, the [CompCogCode](#CompCogCode) shall be shift a bit toward left from low to high; shift out the highest bit and set the *C4array*[*C4ArrayIndex*] as the new lowest one, and update *C4ArrayIndex* accordingly.

In which, C4ArrayIndex is the current cog index, ranging from 0 to 99. *C4array*[*C4ArrayIndex*] is the array of disc codes, "1" meaning conduction and "0" indicating blocked, for details see [REF4].

[iTC\_CC\_ATP-SwRS-0164]

ATP determines the odometer position ready according to the matching of the [CompCogCode](#CompCogCode) and CogCode locked in each interrupt.

In initialization, the *OdometerCogPositonReady* shall be False;

Or else:, if the *OdometerCogPositonReady* was False, then only after the odometer rotated reversely or [WheelFilteredStopped](#WheelFilteredStopped) and re-turned continuous toward the same direction after 8 cogs, ATP shall re-initialize the Counter-Code matching relation and set *OdometerCogPositonReady* as True;

Or else:, if CogCode is different with [CompCogCode](#CompCogCode) in one of interrupt of the cycle, ATP shall set *OdometerCogPositonReady* as False;

Otherwise, ATP keep *OdometerCogPositonReady* unchanging.

[iTC\_CC\_ATP-SwRS-0165]

[TeethCounter](#TeethCounter) used as the odometer cog value in one deferred task, which is the difference of the [CogCounter](#bookmark360) in the last interrupt of adjacent cycle. The calculation of the [TeethCounter](#TeethCounter) shall consider the installation direction of the odometer and the register range of the [CogCounter](#bookmark361).

NOTES:

[TeethCounter](#TeethCounter) is a signed value. If [TeethCounter](#TeethCounter) greater than 0, then means the odometer rotating toward to the train END\_1 direction; other hand, if it less than 0, then means the odometer rotating toward to the END\_2.

[iTC\_CC\_ATP-SwRS-0166]

If a beacon with top-loc received in this cycle, ATP shall record the cog position of the interrupt when and just before the top-loc happen:

[CogPositionBeforeTopLoc](#CogPositionBeforeTopLoc), the [CogCounter](#bookmark366) in the interrupt just before the top-loc happen;

[CogPositionAfterTopLoc](#CogPositionAfterTopLoc), the [CogCounter](#bookmark367) in the interrupt when the top-loc happen.

Which, i means the interrupt received top-loc signal.

[iTC\_CC\_ATP-SwRS-0167]

If all interrupts in one cycle are sensors testing, ATP shall set [SensorTestPerformed](#SensorTestPerformed). Otherwise, set [SensorTestPerformed](#SensorTestPerformed) as False.

[iTC\_CC\_ATP-SwRS-0464]

SensorSequenceDetected\_1，SensorSequenceDetected\_2，SensorSequenceDetected\_3，

If sensors testing performed in this cycle, ATP shall determine the conduction state of each sensor:

If all test results of every interrupts for this sensor are SENSOR\_CONDUCT, ATP shall set sensor sequence detected for this sensor;

Otherwise, does not set this sensor sequence detected.

If sensors testing do not perform, ATP does not set any sensor sequence detected.

[iTC\_CC\_ATP-SwRS-0168]

ATP shall check consistency of sensors whether sensor testing performed or not. Sensors testing result shall declare inconsistent at cycle k ([UnconsistentSensorTest](#UnconsistentSensorTest)) if the following conditions are fulfilled:

Sensors test done at cycle k, and no sequence has been detected on any of the three sensors C1, C2, C3 (SENSOR\_BLOCKED)

Or at cycle k, the expected sequence is detected on all three sensors C1, C2, C3 (SENSOR\_CONDUCT).

Or any of the three sensors is tested as error (SENSOR\_WRONG).

[iTC\_CC\_ATP-SwRS-0171]

Wheel shall consider safely stopped [WheelStopped](#WheelStopped) at cycle k if the following conditions are fulfilled:

sensors test has been performed,

and at least one sensor out of three sensors C1, C2, C3 has detected expected sequence,

and at least one sensor out of three sensors C1, C2, C3 has not detected expected sequence,

and sensors test result combination on three sensors C1, C2, C3 has not changed between cycle k-1 and k.

[iTC\_CC\_ATP-SwRS-0172]

At cycle k, [WheelFilteredStopped](#WheelFilteredStopped) shall change from False to True on raising edge of [WheelStopped](#WheelStopped) information, That is, if:

[WheelStopped](#WheelStopped) information was False at cycle k-1,

and [WheelStopped](#WheelStopped) information was True at cycle k.

and then:

*LastStopCogPosition* is assigned to [TeethCounter](#TeethCounter),

At cycle k, [WheelFilteredStopped](#WheelFilteredStopped) shall change from True to False, according following expression:

the cog moved more than one cog;

[iTC\_CC\_ATP-SwRS-0173]

MaxCountCogsRunInCycleExceeded

ATP shall detect whether the cog number counted in adjacent interrupt is greater than the default maximum cog number on cycle or on interrupt.

[iTC\_CC\_ATP-SwRS-0174]

WheelKinematicsInvalidForCogCount

If the calculated movement exceeds the default one, ATP shall set the wheel kinematics invalid.

* + 1. Outputs
  1. F23-Manage the Odometer State
     1. Inputs
     2. Locals
     3. Processing

In the main task of each cycle, ATP need to estimate the odometer state and calculate WheelMaximumMovement和WheelMinimumMovement, based on the sensor test results and the matching status between cog count and cog number of odometer. As shown in [Figure 57](#Ref2798423931) there are several odometer state:

NOT\_INITIALIZED, just after ATP powered up, the odometer is not initialized. At this moment, ATP should overvalue the wheel displacement to ensure the safety only when the sensor detected that the wheel is completely static.

WAITING\_COG\_POSITION\_CODE\_READY, the odometer starts rolling and continues to roll eight cogs. During this process, ATP should over-estimate wheel displacement to ensure the safety.

INITIALIZED, when odometer has initialized, it enters into the normal working status. If ATP detected the cog count and cog number is matching or the wheel is static, it can calculate the wheel displacement by using the odometer value, otherwise, it should over- estimate the displacement.

INVALID, the odometer is in the invalid status.

* + - 1. State “NOT\_INITIALIZED”

[iTC\_CC\_ATP-SwRS-0175]

From power-up, ATP shall consider that [OdometerState](#OdometerState) is "NOT\_INITIALIZED ".

[iTC\_CC\_ATP-SwRS-0176]

If [OdometerState](#OdometerState) is NOT\_INITIALIZED at cycle k, and if wheel detected stopped at cycle k, then [WheelMinimumMovement](#WheelMinimumMovement) and [WheelMaximumMovement](#WheelMaximumMovement) shall be set to zero.

[iTC\_CC\_ATP-SwRS-0177]

At cycle k, ATP shall consider that [OdometerState](#OdometerState) changes from NOT\_INITIALIZED to WAITING\_COG\_POSITION\_CODE\_READY if:

a falling edge is detected on [WheelFilteredStopped](#WheelFilteredStopped)information,

and sensors test is consistent at cycle k and was consistent at cycle k-1,

[iTC\_CC\_ATP-SwRS-0178]

At cycle k, ATP shall consider that [OdometerState](#OdometerState) changes from NOT\_INITIALIZED to INVALID

if sensors test is not consistent at cycle k.

* + - 1. State “WAITING\_COG\_POSITION\_CODE\_READY”

[iTC\_CC\_ATP-SwRS-0179]

InitializationTimer

ATP shall accumulate the time for waiting cog position ready state.

[iTC\_CC\_ATP-SwRS-0180]

At cycle k, ATP shall consider that [OdometerState](#OdometerState) changes from WAITING\_COG\_POSITION\_CODE\_READY to NOT\_INITIALIZED if:

wheel is detected stopped ([WheelFilteredStopped](#WheelFilteredStopped)),

and cog position remains unknown (not [OdometerCogPositionReady](#OdometerCogPositionReady)),

and there is no sensors test inconsistency,

and time elapsed since last time [OdometerState](#OdometerState) was NOT\_INITIALIZED ([InitializationTimer](#InitializationTimer)) is strictly less than [ATPsetting](#ATPsetting).OdoInitTimeout

[iTC\_CC\_ATP-SwRS-0181]

At cycle k, ATP shall consider that [OdometerState](#OdometerState) changes from WAITING\_COG\_POSITION\_CODE\_READY to INITIALIZED

If:

Cog position is safely known which means that wheel angular position is well-known;

and there is no sensors test inconsistency;

and time elapsed since last time [OdometerState](#OdometerState) was NOT\_INITIALIZED ([InitializationTimer](#InitializationTimer)) is strictly less than [ATPsetting](#ATPsetting).OdoInitTimeout.

[iTC\_CC\_ATP-SwRS-0182]

At cycle k, ATP shall consider that [OdometerState](#OdometerState) changes from WAITING\_COG\_POSITION\_CODE\_READY to INVALID if:

sensors test inconsistency is detected,

or time elapsed since last time [OdometerState](#OdometerState) was NOT\_INITIALIZED ([InitializationTimer](#InitializationTimer)) is more than or equal to the [ATPsetting](#ATPsetting).OdoInitTimeout

[iTC\_CC\_ATP-SwRS-0183]

When odometer is initializing, wheel movement shall be over and under estimated considering maximum acceleration per cycle according with the train front:

* + - 1. State “INITIALIZED”

[iTC\_CC\_ATP-SwRS-0186]

At cycle k, ATP shall consider that [OdometerState](#OdometerState) changes from INITIALIZED to INVALID if:

[OdometerState](#OdometerState) was evaluated Initialized at cycle k-1,

And:

Sensors test result is inconsistent;

Or neither wheel filtered stopped nor cog position ready.

NOTES：

Wheel angular movement is the rotation movement observed on the wheel in cog count from cycle k-1 to k. Angular movement of the wheel can convert into a linear movement by taking into consideration of the uncertainty on wheel diameter measurement provided by calibration process. [WheelMinimumMovement](#WheelMinimumMovement) and [WheelMaximumMovement](#WheelMaximumMovement) represents respectively minimum and maximum curvilinear distance ran between cycle k-1and k by a reference point of the wheel located on the rolling circumference.

[iTC\_CC\_ATP-SwRS-0187]

If motion and speed are available at cycle k, then wheel curvilinear movement calculates as follows:

NOTES:

When odometer cog-counter-code matched, regardless of ATP in END\_1 or END\_2, and no matter the activation of train front, the direction towards END\_1 is always be set as the positive direction. That is, when the movement is greater than 0, indicating the direction of the train is running to END\_1, and vice versa to END\_2.

* + - 1. State “INVALID”

[iTC\_CC\_ATP-SwRS-0189]

At cycle k, ATP shall consider that [OdometerState](#OdometerState) changes from INVALID to NOT\_INITIALIZED if:

[OdometerState](#OdometerState) was evaluated Invalid at cycle k-1,

and wheel is detected stopped ([WheelFilteredStopped](#WheelFilteredStopped)),

and there is no sensors test inconsistency.

[iTC\_CC\_ATP-SwRS-0578]

In invalid status, ATP shall calculate wheel movement by using measured value of the odometer.

* + - 1. Wheel kinematic calculation

[iTC\_CC\_ATP-SwRS-0170]

ATP shall invalidate wheel kinematic if minimum odometer motion is null and sensors test is not performed for more than [ATPsetting](#ATPsetting).OdoTestContradictionDuration.

[iTC\_CC\_ATP-SwRS-0190]

ValidWheelKinematic

Wheel kinematic is valid if odometer is valid, the calculated motion is not greater than the default value, and there is communication with odometer.

[iTC\_CC\_ATP-SwRS-0636]

WheelMinSpeed

[iTC\_CC\_ATP-SwRS-0204]

WheelMaxSpeed

ATP calculates the maximum wheel speed according to the maximum wheel movement; this value is non-negative and rounded up.

* + 1. Outputs
  1. F24-Compensate Sliding Slipping Effect
     1. Inputs
     2. Locals
     3. Processing

As the coded odometer installed on the train brake axle, there is maybe sliding phenomenon between rail and wheel when the train braking. So it will lead to the situation that the detected wheel movement is less than the actual train movement (i.e. the rolling speed of wheel is slower than the actual train speed). In order to avoid this situation, according to the variables of acceleration tested by odometer, ATP will over-estimate the movement and get the over-estimated actual train movement to estimate the train speed for safety consideration.

[iTC\_CC\_ATP-SwRS-0201]

InstantaneousWheelAcceleration

When wheel motion and acceleration are measurable, then instantaneous acceleration computed according following expression:

[iTC\_CC\_ATP-SwRS-0202]

[FilteredWheelAcceleration](#FilteredWheelAcceleration) measurement is the average of [InstantaneousWheelAcceleration](#InstantaneousWheelAcceleration) over FILTERED\_ACCELERATION\_NB cycles for filtering the fluctuation causing by the sampling period.

[iTC\_CC\_ATP-SwRS-0203]

AverageWheelAcceleration

When wheel motion and acceleration are measurable, sliding average acceleration at cycle k defined by following expression:

* + - 1. Sliding state management

[iTC\_CC\_ATP-SwRS-0205]

StartBrakingMovementMax

ATP records the maximum movement when the state transferring from COASTING to BRAKING or SLIDING, or from BRAKING to SLIDING.

[iTC\_CC\_ATP-SwRS-0226]

MaxMotionDuringBrakingOrSliding

ATP shall record the reversed motions during the braking or sliding state.

[iTC\_CC\_ATP-SwRS-0227]

MaxMotionOdometerSignChanged

If the reversed motion during braking or sliding state is greater than a project defined distance, ATP shall consider the motion sign changed.

[iTC\_CC\_ATP-SwRS-0206]

StartSlidingSpeed

ATP shall record the speed when the train begins to slide.

[iTC\_CC\_ATP-SwRS-0207]

TimeInSliding

ATP shall record how many cycles staying in SLIDING state.

[iTC\_CC\_ATP-SwRS-0200]

ATP software shall use the over-estimation model for train movement provided by [Figure 58](#Ref3835323941) state-diagram. The maximum and minimum train motion shall overestimate based on different state as follows:

COASTING. There is not sliding effect during on train coasting or motoring, so ATP need not to overestimate train motion.

BRAKING. When train brakes, which means the measured acceleration is less than the [ATPsetting](#ATPsetting).BrakingStartAcc (normally -0.3m/s^2), ATP shall overestimated the maximum train motion 15% at most.

SLIDING, When the measured acceleration is less than the [ATPsetting](#ATPsetting).SlidingStartAcc (-2.5m/s^2 normally) or the average acceleration is less than [ATPsetting](#ATPsetting).BrakingStartAcc, ATP shall use the train motion before SLIDING as the current train motion.

SKIDDING, If train slides or slips excessively, ATP shall consider odometer motion untrustworthy.

[iTC\_CC\_ATP-SwRS-0209]

The state transfers from “COASTING” to “BRAKING” when:

[iTC\_CC\_ATP-SwRS-0210]

The [MotionOverEstimationState](#MotionOverEstimationState) transfers from “COASTING” to “SLIDING” when:

[iTC\_CC\_ATP-SwRS-0213]

The [MotionOverEstimationState](#MotionOverEstimationState) transfers from “BRAKING” to “SLIDING” when:

[iTC\_CC\_ATP-SwRS-0214]

The [MotionOverEstimationState](#MotionOverEstimationState) transfers from “BRAKING” to “COASTING” when:

[iTC\_CC\_ATP-SwRS-0450]

The [MotionOverEstimationState](#MotionOverEstimationState) transfers from “SLIDING” to “COASTING” when:

NOTES：

The principle of overestimation in SLIDING state based on the following assumptions:

Because the threshold deceleration (the typical value is -2.3m/s^2) used to detect sliding is far less than the full service braking deceleration (normally -1m/s^2), it is not possible to reach the SLIDING state unless the emergency brake applied. The project guaranteed that the absolute value of the minimum emergency brake deceleration is greater than the acceleration due to track gradient, and the rolling stock must cut off the traction during EB applied. Therefore, during SLIDING state, the real speed of the train must gradually reduce. However, if the measured absolute value of the wheel deceleration is too large or too long in the sliding conditions, ATP shall consider the overestimation algorithm as failure.

[iTC\_CC\_ATP-SwRS-0451]

At cycle k, if motion overestimation status is SLIDING, ATP shall consider that sliding effect is ended ([SlidingEnded](#SlidingEnded)) if [FilteredWheelAcceleration](#FilteredWheelAcceleration) is strictly less than [ATPsetting](#ATPsetting).SlippingStopAcc and strictly greater than [ATPsetting](#ATPsetting).SlidingStopAcc for more than [ATPsetting](#ATPsetting).SlidingGripRecoveryTime.

[iTC\_CC\_ATP-SwRS-0218]

The [MotionOverEstimationState](#MotionOverEstimationState) transfers from “SLIDING” to “BRAKING” when:

[iTC\_CC\_ATP-SwRS-0452]

At cycle k, if motion overestimation status is SLIDING, ATP shall consider that sliding is excess ([SlidingExcess](#SlidingExcess)) if [FilteredWheelAcceleration](#FilteredWheelAcceleration) is strictly less than [ATPsetting](#ATPsetting).SlippingStopAcc and strictly greater than [ATPsetting](#ATPsetting).SlidingStopAcc for more than [ATPsetting](#ATPsetting).SlidingExcessTime.

[iTC\_CC\_ATP-SwRS-0217]

The [MotionOverEstimationState](#MotionOverEstimationState) transfers from SLIDING to SKIDDING when:

NOTES：

Because the application of anti-lock braking system for the train of current project, makes the brake acceleration cannot continuous less than [ATPsetting](#ATPsetting).SlidingStopAcc for serious cycles. Therefore, if there was an unexpected instantaneous acceleration less than [ATPsetting](#ATPsetting).SlidingStopAcc, and the after cycles' acceleration met the criteria of [SlidingExcess](#SlidingExcess), then the following two conditions can hold simultaneously.

[iTC\_CC\_ATP-SwRS-0220]

The [MotionOverEstimationState](#MotionOverEstimationState) transfers from SKIDDING to COASTING when:

[iTC\_CC\_ATP-SwRS-0735]

StartBrakingMovementMin

ATP records the minimum movement when the state transferring from COASTING to BRAKING or SLIDING, or from BRAKING to SLIDING.

[iTC\_CC\_ATP-SwRS-0787]

OverestimatedMotionMin

[iTC\_CC\_ATP-SwRS-0788]

OverestimatedMotionMax

In BRAKING state, the maximum train motion overestimated as [ATPsetting](#ATPsetting).SlidingCoefficient (15% normally) at most. If the overestimated motion has greater than the start braking movement, ATP shall use the start breaking movement as the current train motion. That said the train speed during braking could not faster than before.

In state SLIDING:

If both [WheelMaximumMovement](#WheelMaximumMovement) and [StartBrakingMovementMax](#StartBrakingMovementMax) are same direction, and the absolute value of the former is greater than the absolute value of the latter, indicating that due to the sampling error makes the measured movement greater than [StartBrakingMovementMax](#StartBrakingMovementMax). In this case, ATP shall uses [WheelMaximumMovement](#WheelMaximumMovement) as current train maximum motion.

Otherwise, uses [StartBrakingMovementMax](#StartBrakingMovementMax) as train maximum motion.

In other state (COASTING, SKIDDING), uses measured wheel maximum movement as current overestimated maximum train motion.

* + - 1. Slipping state management

[iTC\_CC\_ATP-SwRS-0769]

StartSlippingSpeed

ATP shall record the speed when the train begins to slip.

[iTC\_CC\_ATP-SwRS-0770]

TimeInSlipping

ATP shall record how many cycles staying in SLIPPING state.

[iTC\_CC\_ATP-SwRS-0793]

OdometerAxleMotorized

If the project that odometer installed on the traction axle of the train, ATP shall consider the slipping effect to impact the underestimation of measured wheel movement.

[iTC\_CC\_ATP-SwRS-0199]

ATP software shall use the over-estimation model for train movement provided by [Figure 59](#Ref3835324301) state-diagram. The maximum and minimum train motion shall overestimate based on different state as follows:

COASTING. There is not sliding effect during on train coasting or motoring, so ATP need not to overestimate train motion.

MOTORING, normal traction state (only consdering odometer installed on motorized axle).

SLIPPING, wheel slipping happen (only consdering odometer installed on motorized axle).

SKIDDING, If train slides or slips excessively, ATP shall consider odometer motion untrustworthy.

[iTC\_CC\_ATP-SwRS-0771]

The [MotionUnderEstimationState](#MotionUnderEstimationState) transfers from “COASTING” to “MOTORING” when:

[iTC\_CC\_ATP-SwRS-0772]

The [MotionUnderEstimationState](#MotionUnderEstimationState) transfers from “COASTING” to “SLIPPING” when:

[iTC\_CC\_ATP-SwRS-0773]

The [MotionUnderEstimationState](#MotionUnderEstimationState) transfers from “MOTORING” to “SLIPPING” when:

[iTC\_CC\_ATP-SwRS-0774]

The [MotionUnderEstimationState](#MotionUnderEstimationState) transfers from MOTORING to COASTING” when:

[iTC\_CC\_ATP-SwRS-0775]

The [MotionUnderEstimationState](#MotionUnderEstimationState) transfers from SLIPPING to COASTING when:

[iTC\_CC\_ATP-SwRS-0776]

SlippingEnded

[iTC\_CC\_ATP-SwRS-0777]

The [MotionUnderEstimationState](#MotionUnderEstimationState) transfers from SLIPPING to MOTORING when:

[iTC\_CC\_ATP-SwRS-0778]

SlippingExcess

[iTC\_CC\_ATP-SwRS-0779]

The [MotionUnderEstimationState](#MotionUnderEstimationState) transfers from SLIPPING to SKIDDING when:

[iTC\_CC\_ATP-SwRS-0794]

The [MotionUnderEstimationState](#MotionUnderEstimationState) transfers from SKIDDING to COASTING when:

[iTC\_CC\_ATP-SwRS-0780]

StartMotoringMovementMin

ATP records the minimum movement when the state transferring from COASTING to MOTORING or SLIPPING, or from MOTORING to SLIPPING.

[iTC\_CC\_ATP-SwRS-0795]

UnderestimatedMotionMin

[iTC\_CC\_ATP-SwRS-0796]

UnderestimatedMotionMax

* + - 1. Slide slip modelling validity

[iTC\_CC\_ATP-SwRS-0191]

SlipSlideDetected

For calibration validation purpose, ATP shall consider that slip/side detected if:

motion overestimation modeling status is not coasting nor braking,

or motion underestimation modeling status is not coasting nor motoring.

[iTC\_CC\_ATP-SwRS-0228]

SlipSlideModellingFault

When the overestimation or underesimation state is SKIDDING, or the motion signed changed in BRAKING or SLIDING state, ATP shall consider the overestimation model as fault.

[iTC\_CC\_ATP-SwRS-0229]

ValidSlipSlideModelling

If overestimation model was fault, then ATP considers the model invalid.

[iTC\_CC\_ATP-SwRS-0797]

MaximumSScompensatedMotion

[iTC\_CC\_ATP-SwRS-0798]

MinimumSScompensatedMotion

* + 1. Outputs
  1. F25-Calculate Radar Speed
     1. Inputs
     2. Locals
     3. Processing

[iTC\_CC\_ATP-SwRS-0781]

RadarInfo

[iTC\_CC\_ATP-SwRS-0782]

RadarRawSpeed

[iTC\_CC\_ATP-SwRS-0783]

RadarDirection

[iTC\_CC\_ATP-SwRS-0784]

RadarMotionMax

[iTC\_CC\_ATP-SwRS-0785]

RadarMotionMin

[iTC\_CC\_ATP-SwRS-0786]

RadarSpeedValid

* + 1. Outputs
  1. F26-Detect Odometer Axle Lock
     1. Inputs
     2. Locals
     3. Processing

As ATP calculate the movement and speed only based on the axle odometer assembled, it is necessary to avoid the situation that this axle is locked during the train braking, which may cause the danger that ATP judge the train status as stillness by mistake. Therefore, ATP need to adopt two independent sources of reference speed i.e. when the ATP detected the train is still through the odometer, while the referenced speed showed that the train is running, ATP can conclude that the wheel with odometer is blocked. The [OdometerRef\_1](#OdometerRef_1) and [OdometerRef\_2](#OdometerRef_2) are coming from the speed measuring system of the train and the odometer of the remote ATP respectively. The reference speed structure shows as below:

* + - 1. Odometer speed availability

[iTC\_CC\_ATP-SwRS-0637]

OdometerSpeedAvailable

[iTC\_CC\_ATP-SwRS-0230]

OdometerSpeedUnderThreshold

ATP shall detect whether the measured wheel speed is under threshold.

[iTC\_CC\_ATP-SwRS-0128]

[ReferenceSpeedUnderThreshold\_1](#ReferenceSpeedUnderThreshold_1) defines whether the referenced speed 1 from CCNV is lower than a configurable threshold.

[iTC\_CC\_ATP-SwRS-0129]

[ReferenceSpeedAvailable\_1](#ReferenceSpeedAvailable_1) defines whether the referenced speed 1 from CCNV is valid or not.

[iTC\_CC\_ATP-SwRS-0130]

[ReferenceSpeedUnderThreshold\_2](#ReferenceSpeedUnderThreshold_2) defines whether the referenced speed 2 from CCNV is lower than a configurable threshold.

[iTC\_CC\_ATP-SwRS-0131]

[ReferenceSpeedAvailable\_2](#ReferenceSpeedAvailable_2)shows whether the referenced speed 2 from CCNV is effective or not.

[iTC\_CC\_ATP-SwRS-0231]

[OdometerRef\_1](#OdometerRef_1).[PossiblyDisabled](#bookmark392)

The independent source of odometry reference 1 said to disable if following conditions reached:

local source of odometry is available ([ValidWheelKinematic](#ValidWheelKinematic)),

and [OdometerSpeedUnderThreshold](#OdometerSpeedUnderThreshold) indicates that train speed is greater than reference speed threshold,

and source of odometry reference 1 is available,

and odometer reference 1 indicates that train speed is less than reference speed threshold.

[iTC\_CC\_ATP-SwRS-0232]

[OdometerRef\_2](#OdometerRef_2). [PossiblyDisabled](#bookmark394)

The independent source of odometry reference 2 said to disable if following conditions reached:

local source of odometry is available ([ValidWheelKinematic](#ValidWheelKinematic)),

and [OdometerSpeedUnderThreshold](#OdometerSpeedUnderThreshold) indicates that train speed is greater than reference speed threshold,

and source of odometry reference 2 is available,

and odometer reference 2 indicates that train speed is less than reference speed threshold.

[iTC\_CC\_ATP-SwRS-0233]

[OdometerRef\_1](#OdometerRef_1).[PossiblyEnabled](#bookmark396)

The independent source of odometry reference 1 said to enable if following conditions reached:

local source of odometry is available,

and [OdometerSpeedUnderThreshold](#OdometerSpeedUnderThreshold) indicates that train speed is greater than reference speed threshold,

and source of odometry reference 1 is available,

and odometer reference 1 indicates that train speed is greater than reference speed threshold.

[iTC\_CC\_ATP-SwRS-0234]

[OdometerRef\_2](#OdometerRef_2).[PossiblyEnabled](#bookmark398)

The independent source of odometry reference 2 said to enable if following conditions reached:

local source of odometry is available,

and [OdometerSpeedUnderThreshold](#OdometerSpeedUnderThreshold) indicates that train speed is greater than reference speed threshold,

and source of odometry reference 2 is available,

and odometer reference 2 indicates that train speed is greater than reference speed threshold.

[iTC\_CC\_ATP-SwRS-0235]

[OdometerRef\_1](#OdometerRef_1).[OutOfOrder](#bookmark400)

The independent source of odometry reference 1 is said to be out of order if it is possibly disabled during more than [ATPsetting](#ATPsetting).OdoLockedAxleDisablingLatency.

When the independent source of odometry reference 1 had out of order, it considered not out of order one if the source of odometry reference 1 is possibly enabled during more than [ATPsetting](#ATPsetting).OdoLockedAxleEnablingLatency:

[iTC\_CC\_ATP-SwRS-0236]

[OdometerRef\_2](#OdometerRef_2).[OutOfOrder](#bookmark408)

The independent source of odometry reference 2 is said to be out of order if it is possibly disabled during more than [ATPsetting](#ATPsetting).OdoLockedAxleDisablingLatency.

When the independent source of odometry reference 2 had out of order, It considered not out of order one if the source of odometry reference 2 is possibly enabled during more than [ATPsetting](#ATPsetting).OdoLockedAxleEnablingLatency:

[iTC\_CC\_ATP-SwRS-0237]

[OdometerRef\_1](#OdometerRef_1).[Contradictory](#bookmark416)

The source of odometry reference 1 said to be contradictory with local source of odometry if:

local source of odometry is available ([ValidWheelKinematic](#ValidWheelKinematic))

and [OdometerSpeedUnderThreshold](#OdometerSpeedUnderThreshold) indicates that wheel speed is less than reference speed threshold,

and source of odometry reference 1 is available and not out of order,

and odometer reference 1 indicates that train speed is greater than reference speed threshold.

[iTC\_CC\_ATP-SwRS-0238]

[OdometerRef\_2](#OdometerRef_2). [Contradictory](#bookmark418)

The source of odometry reference 2 said to be contradictory with local source of odometry if:

local source of odometry is available ([ValidWheelKinematic](#ValidWheelKinematic))

and [OdometerSpeedUnderThreshold](#OdometerSpeedUnderThreshold) indicates that wheel speed is less than reference speed threshold,

and source of odometry reference 2 is available and not out of order,

and odometer reference 2 indicates that train speed is greater than reference speed threshold.

* + - 1. Axle locked detection

[iTC\_CC\_ATP-SwRS-0239]

AxlePossiblyLocked

Odometer axle shall consider possibly locked if:

Both independent sources of odometry indicates a contradiction with local odometer,

Or one source of odometry is contradictory and the other one is out of order (or not available).

[iTC\_CC\_ATP-SwRS-0240]

If [AxlePossiblyLocked](#AxlePossiblyLocked) situation lasts more than [ATPsetting](#ATPsetting).OdoLockedAxleTimeout, the odometer axle shall be considered locked. Once [UnrecoverableAxleLocked](#UnrecoverableAxleLocked) set to True, it will stay at state True unless ATP re-initialized.

[iTC\_CC\_ATP-SwRS-0241]

AxleLockedDetectionAvailable

If only one or no source of odometry is available, then ATP shall invalidate kinematic while this situation lasting.

[iTC\_CC\_ATP-SwRS-0242]

WheelTrainKinematicCorrelation

Wheel and train kinematic shall consider correctly correlated if and only if:

odometer axle is not detected locked,

and odometer axle detection is available

* + 1. Outputs
  1. F27-Compute Train Kinematics
     1. Inputs
     2. Locals
     3. Processing

After the overestimation of the sliding effect and consideration of wheel braking decocted by odometers, ATP can calculate the train kinematic parameters which will used by other functions.

[iTC\_CC\_ATP-SwRS-0243]

ValidTrainKinematic

Train kinematic information shall declare invalid if at least one of following condition is True:

odometer kinematic is not valid,

or neither radar speed nor over-estimation modeling of train movement is valid,

or train movement and wheel movement has been detected de-correlated.

[iTC\_CC\_ATP-SwRS-0212]

MaximumTrainMotion

[iTC\_CC\_ATP-SwRS-0736]

MinimumTrainMotion

[iTC\_CC\_ATP-SwRS-0244]

*TrainStopped*, train shall consider strictly stopped if and only if:

wheel is detected strictly stopped,

and train kinematic elaboration is valid,

and wheel is not detected sliding.

[iTC\_CC\_ATP-SwRS-0245]

TrainFilteredStopped

Train shall consider stopped with the tolerance of one cog detection if:

wheel is detected at filtered stop,

and train kinematic elaboration is valid,

and wheel is not detected sliding.

[iTC\_CC\_ATP-SwRS-0638]

TrainHasMoved

[iTC\_CC\_ATP-SwRS-0639]

TrainMinSpeed

[iTC\_CC\_ATP-SwRS-0246]

TrainMaxSpeed

According to the matching of odometer cog counter and code, maximum train speed shall computed using the followings expressions:

[iTC\_CC\_ATP-SwRS-0640]

NewBeaconObtained

[iTC\_CC\_ATP-SwRS-0641]

BeaconBeforeLastObtained

[iTC\_CC\_ATP-SwRS-0642]

BeaconLastObtained

[iTC\_CC\_ATP-SwRS-0247]

If a valid beacon with top-loc received between cycle k-1 and k, then minimum and maximum distance ran since top-loc shall evaluate as the difference between current teeth counter and recorded cog position just before or after top-loc. If there is no beacon received, ATP updates distances from last beacon using the train movements.

[iTC\_CC\_ATP-SwRS-0635]

End2RunningForward

[iTC\_CC\_ATP-SwRS-0758]

End1RunningForward

* + 1. Outputs
  1. F28-Calibrate Wheel Movement

The coding odometer, assembled on axle, has 100 cogs evenly distributed in the outer ring. According to the cog impulse detected by sensor from coded odometer, ATP will estimate the train movement. Therefore, ATP needs to calibrate the cog length by the adjacent MTIB and a verification RB on the track.

* + 1. Inputs
    2. Locals
    3. Processing

There are 100 cogs evenly distributed in the outer ring of odometer, and each cog’s length represented the 1/100 of wheel perimeter, so we can calculate the cog’s length by the wheel diameter. Theoretically, the biggest and the smallest wheel diameter is corresponded to the cog length. When ATP is powered up, it will calculate the train displacement according to the theoretical cog length. When the train passed the fixed distance (this distance described in the track map), ATP will calibrate the real cog length according to the beacon of this fixed distance, and verify this calibration through the third beacon. If the calculation is correct, ATP will use this cog length for the kinematic calculation.

[iTC\_CC\_ATP-SwRS-0192]

As shown in [Figure 510](#Ref3450643131), the odometer calibration state divides into the following four states:

CALI\_WAITING, the state of not calibrated or calibration has failed, waiting for read the MTIB1 beacon.

CALI\_MEASURING, after read MTIB1 and waiting for read MTIB2.

CALI\_VALIDATING, after read MTIB2 and waiting for the verifying RB.

CALI\_COMPLETED, calibration has been verified by the RB successfully.

The following variables shall update during calibration:

[MaxCogCalibration](#MaxCogCalibration), the overestimated calibration for each cog;

[MinCogCalibration](#MinCogCalibration), the underestimated calibration for each cog;

[CalibrationState](#CalibrationState), the state of calibration.

* + - 1. State “CALI\_WAITING”

[iTC\_CC\_ATP-SwRS-0193]

From power-up and while calibration process is not successfully performed, ATP shall use default calibration to compute train motion and shall consider itself in the state of waiting for the first beacon belonging to a couple of calibration.

[iTC\_CC\_ATP-SwRS-0463]

If ATP is in the state of CALI\_WAITING, can transform to the measuring state if following conditions fulfilled:

a valid beacon has been received and this beacon belongs to a couple of calibration,

and train kinematic was valid,

and no excessive slip/slide effect was detected,

and [WheelMinimumMovement](#WheelMinimumMovement) is not null,

Then, ATP shall memorize:

position of the wheel before and after top location signal of received beacon

the ID of received beacon,

the sign of the movement when crossing beacon,

and shall consider itself as CALI\_MEASURING.

* + - 1. State “CALI\_MEASURING”

[iTC\_CC\_ATP-SwRS-0194]

If ATP is in the state of CALI\_MEASURING and following conditions fulfilled:

a valid beacon has been received and the beacon and first memorized calibration beacon is one of possible dedicated couple of calibration,

and train kinematic was valid,

and no excessive slip/slide effect was detected,

and sign of train motion is still identical to thus detected on first beacon signaling,

Then,

if resulting calibration range is included in default calibration range, then ATP shall:

memorize:

position of the wheel before and after top location signal of received beacon,

the id of received beacon,

and shall consider itself as CALI\_VALIDATING.

else: ATP shall consider that calibration process has failed and back to CALI\_WAITING.

Among them, the measured maximum and minimum calibration fetches from the offline-generated counter-calibration table, based on the calculated cog counter.

[iTC\_CC\_ATP-SwRS-0195]

If ATP is in the state of CALI\_MEASURING and following conditions fulfilled:

train kinematic has been detected not valid,

or excessive slip/slide effect has been detected,

or [WheelMinimumMovement](#WheelMinimumMovement) sign is in the opposite direction of thus observed on first beacon or becomes null,

or an unexpected beacon has been received. That is, a beacon not belonging calibration measurement couple.

Then, ATP shall abort calibration process and back to CALI\_WAITING.

* + - 1. State “CALI\_VALIDATING”

[iTC\_CC\_ATP-SwRS-0196]

If ATP is in the state of calibration validation in progress and following conditions fulfilled:

train kinematic has been detected not valid,

or excessive slip/slide effect has been detected,

or [WheelMinimumMovement](#WheelMinimumMovement) sign is in the opposite direction of thus observed on first beacon or becomes null,

or an unexpected beacon has been received. That is, a beacon not belonging calibration validation couple.

In such case, then ATP shall consider that calibration process as not sable and so back to CALI\_WAITING.

[iTC\_CC\_ATP-SwRS-0197]

If ATP is in the state of calibration validation in progress and following conditions fulfilled:

a valid beacon has been received and this beacon is one of possible confirmation beacons related to second beacon signaled of calibration measurement,

and train kinematic was valid,

and no excessive slip/slide effect was detected,

and sign of train motion is still identical to thus detected on first beacon signaling,

and [WheelMinimumMovement](#WheelMinimumMovement) is not null.

Then,

if resulting calibration range fully includes the calibration range in track map, then ATP shall:

update ATP minimum and maximum calibration with last calibration computed on beacons,

and shall consider that calibration process is CALI\_COMPLETED.

else: ATP shall consider that calibration process is not usable and so back to CALI\_WATING waiting for new measurement calibration beacons.

* + - 1. State “CALI\_COMPLETED”

[iTC\_CC\_ATP-SwRS-0198]

If the calibration completed, ATP shall not calibrate when new calibrating beacons read.

* + 1. Outputs
  1. F3-Locate the Train on Track Map
  2. F31-Initialize Train Location
     1. Inputs
     2. Locals
     3. Processing
        1. Localization by beacons

[iTC\_CC\_ATP-SwRS-0461]

End2OrientationByBeacon

The orientation of the train END means the UP or DOWN orientation which this END toward to. When a pair of consecutive beacon read, ATP can determine the orientation for each train END according to the direction of these beacons in track map and the direction of train movement.

NOTES**：**

On a balloon loop, a train can run to the reverse direction without having to shunt or even stop, refer to [Figure 513](#Ref3490298341),which means there are inconsistent orientations for two train ENDs (both END\_1 and END\_2 are toward to UP or DOWN).

[iTC\_CC\_ATP-SwRS-0248]

BeaconLocation

If a valid beacon read, ATP shall calculate actual maximum and minimum location of the END\_2 according to the beacon location in track map, the distance from beacon antenna to the END\_2, the distance after top-loc, the orientation of END\_2 and the direction of train movement. The beacon location indicate the actual external location of the END\_2.

[iTC\_CC\_ATP-SwRS-0280]

TrainLocatedOnBeacon

* + - 1. Localization by memorized location

[iTC\_CC\_ATP-SwRS-0597]

MemorizedLocationAuthorized

[iTC\_CC\_ATP-SwRS-0643]

MemorizedLocationEnable

[iTC\_CC\_ATP-SwRS-0644]

MemorizedLocationAvailable

[iTC\_CC\_ATP-SwRS-0645]

MotionSinceMemorizedLocation

[iTC\_CC\_ATP-SwRS-0646]

TrainPresumablyLocalized

* + - 1. Localization by distant ATP location

[iTC\_CC\_ATP-SwRS-0647]

TrainStoppedStartTime

[iTC\_CC\_ATP-SwRS-0648]

TrainLocatedOnOtherATP

* + - 1. Initialize train location

[iTC\_CC\_ATP-SwRS-0649]

TrainInitialLocation

ATP determine the initial train location by redundant ATP, memorized location and beacon location in order. If train delocalized, the train location should be clear.

* + 1. Outputs
  1. F32-Update Train Location
     1. Inputs
     2. Locals
     3. Processing
        1. Location calculating by movement

[iTC\_CC\_ATP-SwRS-0259]

LocationBeforeReloc

If train has localized on the track map, according to the orientation of END\_2, ATP using the maximum and minimum train motion to update the external or internal location of the END\_2.

* + - 1. Delocalization detecting

[iTC\_CC\_ATP-SwRS-0254]

LocationUntravelable

ATP shall determine whether there is an unknown-status point intersecting with the train location.

If the train has localized at the previous cycle, and fulfills one of the following conditions:

There is an unknown-status divergence of convergence located in the range from train tail to train head, then ATP shall set as True;

Or else:, if there is a convergence with reverse route located in the range from train tail to train head, the ATP shall set as True;

Or else, train crossed the boundary of ATC area, shall set as True.

Otherwise, set as False.

[iTC\_CC\_ATP-SwRS-0460]

InverseLocation

[iTC\_CC\_ATP-SwRS-0253]

LocationUncertaintyExceed

ATP shall calculate the uncertain distance between the external and internal locations of train END\_2.

[iTC\_CC\_ATP-SwRS-0650]

TrainRealignmentOnBeacon

[iTC\_CC\_ATP-SwRS-0768]

RealignmentFailed

[iTC\_CC\_ATP-SwRS-0745]

LocationAfterReloc

[iTC\_CC\_ATP-SwRS-0250]

LocPermanentFailure

[iTC\_CC\_ATP-SwRS-0651]

MotionSinceLastReloc

[iTC\_CC\_ATP-SwRS-0652]

CBTCwithoutKnownPath

[iTC\_CC\_ATP-SwRS-0074]

If either of ends can ensure the train integrity, ATP shall set [TrainUnitIntegrity](#TrainUnitIntegrity) as True. If the project is not configured with the capture of train integrity, it is sure that the train can guarantees the integrity.

[iTC\_CC\_ATP-SwRS-0278]

LocalizationFault

If the localization state is not NOT\_LOCALIZED, ATP shall determine whether the localization fault happens or not, according to the following pseudo-codes:

* + - 1. Train location calculating

[iTC\_CC\_ATP-SwRS-0283]

TrainLocalized

Only the localization state is LOCALIZED, ATP shall consider the train has localized.

When the train passed the continuous two beacons, ATP can judge the initial location and direction according to the position and the sequences of above-mentioned beacons in track map. Later, ATP can update the train location in the track map in each cycle based on the train movement combined with sliding overestimation and wheel block consideration. If ATP received a new beacon, it will realign the train location according to this beacon. For safety, ATP needs to maintain the location information from the external and internal side of each train

end.

[iTC\_CC\_ATP-SwRS-0258]

TrainLocation

* + 1. Outputs
  1. F33-Confirm Train Localization
     1. Inputs
     2. Locals
     3. Processing
        1. Determine the train front

ATP shall determine the train front by cab activation from project configured VIOM input port. If the driver did not activate either end of the cab, ATP shall judge the train front according to the info from CCNV.

[iTC\_CC\_ATP-SwRS-0076]

ATP shall consider the driver is in this cab if it captures that either end of cab activated, which shown by the data from [DriverInCab\_1](#DriverInCab_1) or [DriverInCab\_2](#DriverInCab_2).

[iTC\_CC\_ATP-SwRS-0139]

DriverInTrain

If the active status is different between two ENDs of the train, ATP consider there is a driver in train.

[iTC\_CC\_ATP-SwRS-0127]

[NonVitalSelectedFrontEnd](#NonVitalSelectedFrontEnd) represents the train front choice from CCNV.

[iTC\_CC\_ATP-SwRS-0138]

TrainFrontEnd

If there is a driver in the train, the train front is the activated END. or else: the front determined by CCNV.

Otherwise, the train front is the default one or the front one when train is moving.

[iTC\_CC\_ATP-SwRS-0281]

TrainFrontOrientation

The train front orientation is the orientation of the active train END.

[iTC\_CC\_ATP-SwRS-0249]

TrainFrontLocation

ATP updates the train front location according to the active train END.

[iTC\_CC\_ATP-SwRS-0255]

TrainRearLocation

ATP updates the train rear locations according to the active train END.

* + - 1. Location path known status

[iTC\_CC\_ATP-SwRS-0757]

LocationPathKnown

In BM mode, if train localized but EOA is invalid, [LocationPathKnown](#LocationPathKnown) cannot be TRUE when train cross switch. After train crossed switch and relocalized by beacon successfully, [LocationPathKnown](#LocationPathKnown) can be set to TRUE.

[iTC\_CC\_ATP-SwRS-0653]

TrainLocatedOnKnownPath*，*[LocationPathKnown](#LocationPathKnown)。

[iTC\_CC\_ATP-SwRS-0275]

LocalizationState

* + - 1. Location confirmation for ZC

[iTC\_CC\_ATP-SwRS-0654]

MemLocationNotConfirmed

[iTC\_CC\_ATP-SwRS-0655]

LocationUncertaintyExceedTime

[iTC\_CC\_ATP-SwRS-0656]

LocationNotUncertaintyExceed

[iTC\_CC\_ATP-SwRS-0411]

LocalizedAuthorizationForSweepping

ATP shall send the current localization status to the [ZC](#bookmark480).

[iTC\_CC\_ATP-SwRS-0412]

TrainConfirmedLocalized

ATP shall send the status of the localization status whether confirmed.

* + - 1. Memorizing train location

[iTC\_CC\_ATP-SwRS-0657]

TrainIncludedInSleepingZone

[iTC\_CC\_ATP-SwRS-0659]

WritingMemLocRequest

Only when train has moved and filtered stopped in sleeping zone, can ATP writing memorized location information.

[iTC\_CC\_ATP-SwRS-0660]

ClearingMemLocRequest

When train has moved and does not fulfill the condition of writing memory location, ATP shall clear memorized location information.

[iTC\_CC\_ATP-SwRS-0661]

MemLocWritten

* + 1. Outputs
  1. F4-Monitor Train Energy
  2. F41-Determine the EOA
     1. Inputs
     2. Locals
     3. Processing

In the block model and CBTC model ATP have different processing mode for EOA. In the CBTC mode, EOA is coming from ZC message. In block mode, according to the BM variant status, ATP searches the target from downstream of the train front location. When it gets the first restricted variant or track border, ATP shall regard it at the EOA for this cycle, and the validity of EOA shall be the same as the BM variants.

* + - 1. Determing EOA in block mode

[iTC\_CC\_ATP-SwRS-0270]

[RestrictiveSignalOverrun](#RestrictiveSignalOverrun), ATP shall determine whether the location of maximum train head overruns a restricted signal in BLOCK mode.

When all of the following conditions fulfilled, ATP considers the train has overrun a restricted signal in this cycle, and set [RestrictiveSignalOverrun](#RestrictiveSignalOverrun) as True.

Train has localized;

And the current type of EOA is BLOCK\_MODE\_EOA;

And [RestrictiveSignalOverrun](#RestrictiveSignalOverrun) was False at the last cycle;

And the moving direction in current cycle is toward on the train front end;

And the maximum location of train front end passes the position of the signal in this cycle;

And the status of the signal is restriction or overlap established.

Otherwise, ATP set [RestrictiveSignalOverrun](#RestrictiveSignalOverrun) as False.

[iTC\_CC\_ATP-SwRS-0662]

TrainInBMinitialZone

[iTC\_CC\_ATP-SwRS-0663]

TrainEnteredInBMinitialZoneAge

[iTC\_CC\_ATP-SwRS-0664]

StopAssuredPointCrossed

[iTC\_CC\_ATP-SwRS-0665]

CBIminProductionAgeSinceSSAcrossing

[iTC\_CC\_ATP-SwRS-0666]

CBIminProductionAge

[iTC\_CC\_ATP-SwRS-0667]

ReceivedVariantsAfterEnteredBMinitialZone

[iTC\_CC\_ATP-SwRS-0504]

BlockModeEOAvalid

* + - 1. Determing EOA in CBTC mode

[iTC\_CC\_ATP-SwRS-0669]

TrainInSMIzone

[iTC\_CC\_ATP-SwRS-0160]

CBTCmodeEOAvalid

[iTC\_CC\_ATP-SwRS-0670]

CBTCmodeEOAlocation

* + - 1. End of authority for train

[iTC\_CC\_ATP-SwRS-0671]

EndOfAuthorityValid

* + - 1. Traction authorized direction

[iTC\_CC\_ATP-SwRS-0141]

TractionAuthorisedSenseEnd1

If current EOA is valid and whose orientation is END\_1, ATP shall authorize the train can move toward END\_1.

[iTC\_CC\_ATP-SwRS-0142]

TractionAuthorisedSenseEnd2

If current EOA is valid and whose orientation is END\_2, ATP shall authorize the train can move toward END\_2.

* + 1. Outputs
  1. F42-Manage Coerced Permissive or Restrictive

5.24.1. Inputs

5.24.2. Locals

5.24.3. Processing

5.24.3.1 Overlap releasable

[iTC\_CC\_ATP-SwRS-0600]

BMoverlapReleasableSendable

[iTC\_CC\_ATP-SwRS-0673]

OverlapReleasable

[iTC\_CC\_ATP-SwRS-0674]

CrossedOverlapTimerInitialSignal

[iTC\_CC\_ATP-SwRS-0675]

OverlapTimer

[iTC\_CC\_ATP-SwRS-0676]

OverlapTimerPermissive

5.24.3.2 Coerced restrictive

[iTC\_CC\_ATP-SwRS-0677]

NotCoercedRestrictive\_1

[iTC\_CC\_ATP-SwRS-0678]

NotCoercedRestrictive\_2

[iTC\_CC\_ATP-SwRS-0679]

NotCoercedRestrictive\_3

[iTC\_CC\_ATP-SwRS-0680]

NotCoercedRestrictive\_4

[iTC\_CC\_ATP-SwRS-0681]

CoercedRestrictive

5.24.3.3 Coerced permissive

[iTC\_CC\_ATP-SwRS-0682]

CoercedPermissive\_1

[iTC\_CC\_ATP-SwRS-0683]

CoercedPermissive\_2

[iTC\_CC\_ATP-SwRS-0684]

CoercedPermissive\_3

[iTC\_CC\_ATP-SwRS-0685]

CoercedPermissive\_4

[iTC\_CC\_ATP-SwRS-0686]

CoercedPermissive

5.24.4 Outputs

* 1. F43-Manage Temporary Speed Restriction

5.25.1. Inputs

5.25.2. Locals

5.25.3. Processing

5.25.3.1. Receiving TSR message

[iTC\_CC\_ATP-SwRS-0687]

TSRreportReceived

[iTC\_CC\_ATP-SwRS-0099]

TSRreportAvailable

[iTC\_CC\_ATP-SwRS-0688]

LastTSRreportAge

5.25.3.2. Parsing TSR

[iTC\_CC\_ATP-SwRS-0102]

ReceivedTSRdatabase

ATP shall map the TSR message received from LC to structure of block. It need to judge whether there is corresponding TSR for each BLOCK in the track map. If yes, ATP shall update the abscissa of the starting and ending points, as well as the restriction speed. During the process, it need to transfer the abscissa and speed unit of TSR message to the corresponding one used in ATP.

NOTES：

For TSR processing, the iTC system supports only one TSR at one BLOCK at most. The beginning and the termination point of the TSR can be set any position in this block. However, it never sustains the situation that there are more than one TSR in one block.

5.25.4. Outputs

* 1. F44-Compute Train Energy

This module calculates the train energy.

* + 1. Inputs
    2. Locals
    3. Processing

Document [[REF10]](#bookmark542) describes the principle and implementation of the train energy monitoring. When the EOA is valid, ATP calculates the maximum energy position, where the traction has cut off and the brake begin to effect, based on the overestimated gradient and train kinematics. According to the principle of conservation of energy, ATP uses the train maximum energy to compare with the restricted energy of the vital zones train located or the downstream constraint points, to determine whether the current train energy exceeded the environment limits. If the over energy was detected, ATP shall request emergency braking to ensure that in the worst conditions, the train will not exceed the vital zone's speed limits or will not overrun the downstream EOA.

[iTC\_CC\_ATP-SwRS-0309]

If EOA is valid, ATP shall calculate the distance and the speed of the train moved after traction cutoff period (image73), according to the current train maximum speed, the maximum acceleration of traction and the maximum acceleration of the gradient.

Where,

image73，the latency from ATP trigger EB command to the traction cut off.

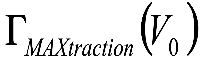
, from [ATPsetting](#ATPsetting), is the maximum acceleration of traction at current speed.

image80, from [TrackMap](#TrackMap), is the maximum acceleration of the gradient in the block start point train front end located plus to the distance (image81)

[iTC\_CC\_ATP-SwRS-0310]

If EOA is valid, ATP shall calculate the distance and the speed of the train moved after the traction cutoff period plus emergency brake applied period (image82), according to the current train maximum speed, the maximum acceleration of traction and the maximum acceleration of the gradient.

Where, image86，the [ATPsetting](#ATPsetting).EBtractionToBrakingLatency latency from RS cut off the traction to the emergency braking applied.

[iTC\_CC\_ATP-SwRS-0312]

TrainEnergy ，ATP shall calculate the train energy where EB indeed applied. The calculation shall consider the kinetic energy and the error of the potential energy.

The [ATPsetting](#ATPsetting).MPauthAltitudeMaxErrorEnergy means an algorithm error caused by offline tool to calculate the compensation gradients.

* + 1. Outputs
  1. F45-Process Singularities

For the localized train, ATP shall consider the singularities within the range of train locations from the head END to the tail. For instance, ATP need to estimate whether the train is in the authorized area for opening the door in station, or whether the train location intersects with the PSD, or whether the train location exceeds the signal with restricted status.

5.27.1. Inputs

5.27.2. Locals

5.27.3. Processing

For the vital zone intersected with the train location, such as PSR, TSR, restricted PSD or Protection Zone etc., ATP shall inhibit the instantaneous velocity of the train will not greater than the limit. In addition, ATP shall request EB if the train location intersected with the EOA, uncontrolled point or track end.

Even in the worst cases, ATP should ensure that the train could stop in the upstream of the constraint point, i.e. not exceeding EOA. Meanwhile, if there is PSR or TSR area in the downstream, ATP also should ensure trains enter the area below the speed restriction. Therefore, ATP needs to calculate kinetic energy and static energy of each potential limitation to ensure that the current train energy is lower than the limitation energy.

5.27.3.1. Train speed limit

[iTC\_CC\_ATP-SwRS-0690]

ZoneVSLNotExceedTrainSpeedLimit

5.27.3.2. PSR

[iTC\_CC\_ATP-SwRS-0691]

ZoneVSLnotExceedPSR

[iTC\_CC\_ATP-SwRS-0692]

PointVSLnotExceedPSR

NOTES:

If the train tail intersected with a PSR (or TSR) area, and the speed of train is lower but the [V2EbApplied](#V2EbApplied) is higher than the limitation. In accordance with the above approach will result in EB, although when the train runs to the EB applied position, the train may have left the PSR area.

5.27.3.3. BSR

[iTC\_CC\_ATP-SwRS-0693]

ZoneVSLnotExceedBSR

[iTC\_CC\_ATP-SwRS-0694]

PointVSLnotExceedBSR

5.27.3.4. TSR

[iTC\_CC\_ATP-SwRS-0069]

According to the status of *TSRcontrollinhibition*, ATP can judge whether it is necessary to handle TSR information.

[iTC\_CC\_ATP-SwRS-0695]

ZoneVSLnotExceedTSR

[iTC\_CC\_ATP-SwRS-0696]

PointVSLnotExceedTSR

5.27.3.5. Track ends

[iTC\_CC\_ATP-SwRS-0697]

ZoneVSLnotExceedOTE

[iTC\_CC\_ATP-SwRS-0698]

PointVSLnotExceedOTE

[iTC\_CC\_ATP-SwRS-0699]

ZoneVSLnotExceedCTE

[iTC\_CC\_ATP-SwRS-0700]

PointVSLnotExceedCTE

5.27.3.6. Signals and overlap ends

[iTC\_CC\_ATP-SwRS-0701]

ZoneVSLnotExceedSignal

[iTC\_CC\_ATP-SwRS-0702]

PointVSLnotExceedSignal

[iTC\_CC\_ATP-SwRS-0703]

ZoneVSLnotExceedOverlap

[iTC\_CC\_ATP-SwRS-0704]

PointVSLnotExceedOverlap

5.27.3.7. Uncontrolled points

[iTC\_CC\_ATP-SwRS-0705]

ZoneVSLnotExceedSwitch

[iTC\_CC\_ATP-SwRS-0706]

PointVSLnotExceedSwitch

5.27.3.8. Protection zones

[iTC\_CC\_ATP-SwRS-0707]

ZoneVSLnotExceedPZ

[iTC\_CC\_ATP-SwRS-0708]

PointVSLnotExceedPZ

5.27.3.9. PSD zones

[iTC\_CC\_ATP-SwRS-0709]

ZoneVSLnotExceedPSD

[iTC\_CC\_ATP-SwRS-0710]

PointVSLnotExceedPSD

5.27.3.10. ZC zones without LC authorization

[iTC\_CC\_ATP-SwRS-0711]

ZoneVSLnotExceedZC

[iTC\_CC\_ATP-SwRS-0712]

PointVSLnotExceedZC

5.27.3.11. EOA

[iTC\_CC\_ATP-SwRS-0713]

ZoneVSLnotExceedEOA

[iTC\_CC\_ATP-SwRS-0714]

PointVSLnotExceedEOA

5.27.3.12. Monitor train energy against restrictive zones or points

[iTC\_CC\_ATP-SwRS-0318]

ZoneVSLNotExceed

ATP shall determine whether train exceeds the vital speed limitation of the restrictive zone, by comparing the energy between the energy of the train and the energy of the zone.

[iTC\_CC\_ATP-SwRS-0324]

PointVSLNotExceed

ATP shall determine whether train exceeds the vital speed limitation of the restrictive point, by comparing the energy between the energy of the train and the kinetic added potential energy of the point.

* + 1. Outputs
  1. F46-Determine Over Energy

5.28.1. Inputs

5.28.2. Locals

5.28.3. Processing

ATP shall trigger EB to make sure that the train can fulfill the limitation requirements of speed or location, no matter the train energy exceeded the energy of train intersected zone or constraint point of the downstream.

5.28.3.1. Motion protection inhibition

[iTC\_CC\_ATP-SwRS-0064]

MotionProtectionInhibition

5.28.3.2. Determing over energy

[iTC\_CC\_ATP-SwRS-0325]

TrainPossiblyInOverEnergy

If the train energy exceeds the zone of point vital speed limitation, ATP shall consider the train possibly over energy.

[iTC\_CC\_ATP-SwRS-0326]

TrainEnergyControlDisabled

If the RMF or RMR mode selected, ATP shall not monitor the train energy.

[iTC\_CC\_ATP-SwRS-0327]

EBforOverEnergy

ATP shall request emergency braking if train is possibly in over-energy and train speed control enabled and if following conditions fulfilled:

the train is not detected at filtered stop,

or the train is detected at filtered stop and:

safe immobilization customization setting for this control indicates to use emergency brake,

or safe immobilization customization setting for this control indicates to use emergency brake when it was already applied.

[iTC\_CC\_ATP-SwRS-0328]

PBforOverEnergy

ATP shall request parking braking if train is possibly in over-energy and train speed control enabled and if following conditions fulfilled:

the train is detected at filtered stop,

and safe immobilization customization setting for this control indicates to use parking brake.

5.28.4. Outputs

* 1. F5-Monitor Train Position and Speed
  2. F51-Moral Time Control

This module monitors whether the train is staying in the fuzzy time area longer than expected.

* + 1. Inputs
    2. Locals
    3. Processing

Because ATP always overestimates the maximum location of the train front end, it is possible to consider the train has crossed a signal but actually not. In this case, if the CI cancelled the route downstream of this signal and it changed to restricted status accordingly, ATP cannot prohibit the train to moving on the unauthorized route (because ATP think the train has crossed the signal and the CI cannot change the route).

To avoid this situation, in Block operation mode, ATP needs to consider an additional area downstream the signal, called moral time area. If the train location intersects with the moral time area more than a configurable period just after the signal change from the permissive to the restrictive status, ATP shall trigger the EB. In CBTC mode, the monitoring of moral time is handled by ZC.

[iTC\_CC\_ATP-SwRS-0292]

NotOnRestrictiveMoralTimeArea\_1

[iTC\_CC\_ATP-SwRS-0293]

NotOnRestrictiveMoralTimeArea\_2

[iTC\_CC\_ATP-SwRS-0294]

If ATP cannot determine train is [NotOnRestrictiveMoralTimeArea\_1](#NotOnRestrictiveMoralTimeArea_1), and if this situation lasts more than [ATPsetting](#ATPsetting). MTtimeout cycles, ATP shall consider that route exclusivity is not guaranteed and [RouteExclusivityGuaranted\_1](#RouteExclusivityGuaranted_1) shall be set to False.

If ATP detects that train is [NotOnRestrictiveMoralTimeArea\_1](#NotOnRestrictiveMoralTimeArea_1), route exclusivity shall consider as guaranteed for that direction of travel and [RouteExclusivityGuaranted\_1](#RouteExclusivityGuaranted_1) shall set to True

[iTC\_CC\_ATP-SwRS-0295]

If ATP cannot determine train is [NotOnRestrictiveMoralTimeArea\_2](#NotOnRestrictiveMoralTimeArea_2), and if this situation lasts more than [ATPsetting](#ATPsetting).MTtimeout cycles, ATP shall consider that route exclusivity is not guaranteed and [RouteExclusivityGuaranted\_2](#RouteExclusivityGuaranted_2) shall set to False.

If ATP detects that train is [NotOnRestrictiveMoralTimeArea\_2](#NotOnRestrictiveMoralTimeArea_2), route exclusivity shall consider as guaranteed for that direction of travel and [RouteExclusivityGuaranted\_2](#RouteExclusivityGuaranted_2) shall set to True

[iTC\_CC\_ATP-SwRS-0296]

HazardousMotionOnNonExclusiveRoute ,If [RouteExclusivityGuaranted\_1](#RouteExclusivityGuaranted_1) is False, ATP shall request emergency braking if and only if:

[TrainFrontEnd](#TrainFrontEnd) is not END\_2,

RM forward nor RM reverse are not selected,

and block mode is not selected.

If [RouteExclusivityGuaranted\_2](#RouteExclusivityGuaranted_2) is False, ATP shall request emergency braking if and only if:

[TrainFrontEnd](#TrainFrontEnd) is not END\_1,

RM forward nor RM reverse driving mode are not selected,

and block mode is not selected.

[iTC\_CC\_ATP-SwRS-0297]

PBonNonExclusiveRoute

ATP shall request parking braking if train considered too near from a non-exclusive route and if following conditions are fulfilled:

the train is detected at filtered stop,

safe immobilization customization setting for this control indicates to use parking brake.

[iTC\_CC\_ATP-SwRS-0298]

EBonNonExclusiveRoute

ATP shall request emergency braking if train considered too near from a non-exclusive route and if following conditions are fulfilled:

the train is not detected at filtered stop,

or the train is detected at filtered stop and:

safe immobilization customization setting for this control indicates to use emergency brake,

or safe immobilization customization setting for this control indicates to use emergency brake when it was already applied.

5.30.4 Outputs

* 1. F52-Prevent Train Moving in Undetectable Danger

This function used to monitor whether the train is running with undetectable danger risk.

* + 1. Inputs
    2. Locals
    3. Processing
       1. Permissive/ not restrictive zone logical inputs

[iTC\_CC\_ATP-SwRS-0717]

PermissiveZoneLogicalInput

[iTC\_CC\_ATP-SwRS-0718]

NotRestrictiveZoneLogicalInput

5.31.3.2. No undetectable dange

ATP shall avoid the train in "undetectable danger risk", which means that the train runs on the direction not authorized by the driver or by CCNV in ATB driving mode. In this case, if the train runs more than a configurable distance, ATP shall trigger emergency brake and keep it for a configurable period.

[iTC\_CC\_ATP-SwRS-0582]

NoUndetectableDanger\_1

The No Undetectable Danger in Extremity 1 shall be consider as permissive status according to project configuration.

[iTC\_CC\_ATP-SwRS-0583]

NoUndetectableDanger\_2

The "No Undetectable Danger in Extremity 2" shall be consider as permissive status according to project configuration.

[iTC\_CC\_ATP-SwRS-0285]

UndetectableDangerRiskForNoNUDE

If there is neither No Undetectable Danger in Extremity 1 nor No Undetectable Danger in Extremity 2 inputs, ATP shall consider the train is possible under the risk of undetectable danger.

[iTC\_CC\_ATP-SwRS-0286]

PBforUndetectableDangerRisk

ATP shall request a parking braking if the possibility of an undetected danger has proven to be and if following conditions are fulfilled:

the train is detected at filtered stop,

safe immobilization customization setting for this control indicates to use parking brake.

[iTC\_CC\_ATP-SwRS-0287]

When the driver does not authorize the train running toward the END\_1, ATP shall accumulate the distance of the train running toward to the END\_1.

If in initialization, or the [NoUndetectableDanger\_1](#NoUndetectableDanger_1) is True, or the train has triggered EB and has stopped, ATP set this distance to 0;

Else if train kinematic has invalid, ATP set this distance to the default value.

Else if the odometer has initialized:

If the [MaximumTrainMotion](#MaximumTrainMotion)is greater than 0, ATP accumulate the maximum movement in this cycle with the distance of last cycle;

Or if the [MaximumTrainMotion](#MaximumTrainMotion)is less than or equal to 0, ATP use the minimum movement of this cycle plus to the distance last cycle (in fact, decrease the accumulated distance). The minimum of this accumulated distance is 0.

Otherwise, keep the distance unchanged.

[iTC\_CC\_ATP-SwRS-0288]

NUDEdistanceAccount\_2

When the driver does not authorize the train running toward the END\_2, ATP shall accumulate the distance of the train running toward to the END\_2.

If in initialization, or the [NoUndetectableDanger\_2](#NoUndetectableDanger_2) is True, or the train has triggered EB and has stopped, ATP set this distance to 0;

Else if train kinematic has invalid, ATP set this distance to the default value.

Else if the odometer has initialized:

if the [MaximumTrainMotion](#MaximumTrainMotion) is less than 0, ATP accumulate the maximum movement in this cycle with the distance of last cycle;

Else: if the [MaximumTrainMotion](#MaximumTrainMotion) is greater than or equal to 0, ATP use the minimum movement of this cycle plus to the distance last cycle (in fact, decrease the accumulated distance). The minimum of this accumulated distance is 0.

Otherwise, keep the distance unchanged.

[iTC\_CC\_ATP-SwRS-0289]

UndetectDangerMotionWithoutNUDE

When the train has moved without NUDE more than project-restricted distance, ATP shall set this value to True.

[iTC\_CC\_ATP-SwRS-0290]

The EB request shall be maintained to True during the application time [ATPsetting](#ATPsetting).NUDEtrainStopDurationBeforeEBrelease, if the train has moved without NUDE more than project restricted distance.

When [UndetectDangerMotionWithoutNUDE](#UndetectDangerMotionWithoutNUDE) is True, ATP shall set [EBappliedForMotionWithoutNUDE](#EBappliedForMotionWithoutNUDE) to True;

When [UndetectDangerMotionWithoutNUDE](#UndetectDangerMotionWithoutNUDE) change from True to False, ATP shall maintain [EBappliedForMotionWithoutNUDE](#EBappliedForMotionWithoutNUDE) to True in period [ATPsetting](#ATPsetting).NUDEtrainStopDurationBeforeEBrelease；

Over the time, set this value to False.

[iTC\_CC\_ATP-SwRS-0291]

EBforUndetectableDangerRisk

When the train has triggered emergency brake causing by the "undetectable danger risk" and has stopped, ATP shall determine whether keeping the EB output according to the project configuration.

5.31.4. Outputs

* 1. F53-Monitor Train Speed
     1. Inputs
     2. Processing
        1. RM mode speed monitoring

If the RM forward or reverse modes selected, ATP shall monitor whether the train is overspeed based on the RM limitation.

[iTC\_CC\_ATP-SwRS-0743]

ConditionForRMlimitSpeed

[iTC\_CC\_ATP-SwRS-0744]

RMlimitSpeedApplied

[iTC\_CC\_ATP-SwRS-0497]

NoDangerForRMoverSpeed

[iTC\_CC\_ATP-SwRS-0734]

EBforRMoverSpeed

5.32.2.2. Monitoring memorized location speed limit

[iTC\_CC\_ATP-SwRS-0751]

NoDangerForMemorizedLocationOverSpeed

[iTC\_CC\_ATP-SwRS-0719]

EBforMemorizedLocationOverSpeed

5.32.3. Outputs

* 1. F54-Monitor Rollback Train Speed
     1. Inputs
     2. Locals
     3. Processing

[iTC\_CC\_ATP-SwRS-0065]

[RMRselectedDrivingMode](#RMRselectedDrivingMode) represents the choice of RMR.

When the train is not on RMR mode, and the train moved backward related to the active cab, called rollback. ATP shall request EB if the speed of the rollback is greater than the project limits. If the rollback distance is greater than the project limits, ATP shall request the permanent EB, which cannot release.

[iTC\_CC\_ATP-SwRS-0300]

When train front extremity is END\_1 and traction effort is supposed to be in the direction of travel, [RollbackDistanceAccount\_1](#RollbackDistanceAccount_1) is the estimated maximum distance which separates current front extremity 1 position to last most forward position reached by this extremity. ATP shall evaluate [RollbackDistanceAccount\_1](#RollbackDistanceAccount_1) in order to control that speed does not exceed [ATPsetting](#ATPsetting).MPnotAuthLimitSpeed .

[iTC\_CC\_ATP-SwRS-0301]

When train front extremity is END\_2 and traction effort is supposed to be in the direction of travel, [RollbackDistanceAccount\_2](#RollbackDistanceAccount_2) is the estimated maximum distance which separates current front extremity 2 position to last most forward position reached by this extremity. ATP shall evaluate [RollbackDistanceAccount\_2](#RollbackDistanceAccount_2) in order to control that speed does not exceed [ATPsetting](#ATPsetting).MPnotAuthLimitSpeed.

[iTC\_CC\_ATP-SwRS-0302]

From ATP power-up, [UnrecoverableRollbackOverSpeed](#UnrecoverableRollbackOverSpeed) shall initialize to False.

[UnrecoverableRollbackOverSpeed](#UnrecoverableRollbackOverSpeed) shall be set to True if and only if following conditions are fulfilled:

driving selector indicates that traction effort is supposed to be in the direction of travel,

train front extremity is END\_2 or END\_1,

and rollback limit speed currently applicable is null for this direction of travel.

Once [UnrecoverableRollbackOverSpeed](#UnrecoverableRollbackOverSpeed) set as True, it shall stay at state True while ATP is not reboot.

[iTC\_CC\_ATP-SwRS-0303]

[RollbackOverSpeed](#RollbackOverSpeed) shall be True if and only if following conditions are fulfilled:

driving selector indicates that traction effort is supposed to be in the direction of travel,

train front extremity is END\_2 or END\_1,

and movement observed is in the opposite direction of travel,

and over-estimated train speed is greater than [ATPsetting](#ATPsetting).MPnotAuthLimitSpeed currently applicable for this direction of travel and rollback speed restrictions is not null.

Or:

train has reached a position due a rollback movement which is unrecoverable,

Or:

train kinematic is invalid,

[iTC\_CC\_ATP-SwRS-0304]

EBforRollbackOverSpeed

ATP shall request emergency braking if a reverse speed limit is over-run for unwilling rollback or excessive reverse motion.

5.33.4. Outputs

* 1. F55-Monitor Reverse Train Speed
     1. Inputs
     2. Locals
     3. Processing

When the train selected on RMR mode, the train moved reversely related to the active cab. ATP shall request EB if the speed of the reverse is greater than the project limits. If the train reversing distance is greater than the project limits, ATP shall keep requesting EB to inhibit train moving on RMR mode.

[iTC\_CC\_ATP-SwRS-0759]

[LongDistanceReverseAuthorized](#LongDistanceReverseAuthorized) represents the authorization of long distance reverse.

[iTC\_CC\_ATP-SwRS-0305]

When train front extremity is END\_1 and traction effort is supposed to be in the opposite direction of travel, [ReverseDistanceAccount\_1](#ReverseDistanceAccount_1) is the estimated maximum distance which separates current front extremity 1 position to last most forward position reached by this extremity. ATP shall evaluate [ReverseDistanceAccount\_1](#ReverseDistanceAccount_1) in order to control that speed does not exceed reverse speed limit function.

[iTC\_CC\_ATP-SwRS-0306]

When train front extremity is END\_2 and traction effort is supposed to be in the opposite direction of travel, [ReverseDistanceAccount\_2](#ReverseDistanceAccount_2) is the estimated maximum distance which separates current front extremity 2 position to last most forward position reached by this extremity. ATP shall evaluate [ReverseDistanceAccount\_2](#ReverseDistanceAccount_2) in order to control that speed does not exceed [ReverseSpeedRestrictions](#ReverseSpeedRestrictions) reverse speed limit function.

[iTC\_CC\_ATP-SwRS-0753]

ReverseSpeedRestrictions

[iTC\_CC\_ATP-SwRS-0307]

[ReverseOverSpeed](#ReverseOverSpeed) shall be True if following conditions fulfilled:

driving selector indicates that traction effort is supposed to be in the opposite direction of travel,

train front extremity is END\_2 or END\_1,

and movement observed is the opposite direction of travel,

and:

over-estimated train speed is greater than reverse speed restrictions currently applicable for this direction of travel,

or else: if reverse speed restrictions currently applicable is null for this direction of travel,

Or else: train kinematic is invalid.

[iTC\_CC\_ATP-SwRS-0308]

EBforReverseOverSpeed

ATP shall request emergency braking if a reverse speed limit is over-run for unwilling rollback or excessive reverse motion.

5.34.4. Outputs

* 1. F6-Protect Passengers Entrance and Exit from the Train

This module describes the requirements for passengers entrance and exist of the train.

* 1. F61-Elaborate Door Opening Authorization

This module is judging the criteria for door opening.

* + 1. Inputs
    2. Locals
    3. Processing

Only when the ATP has sure that the train is completely located in the platform area and the safely immobilized, ATP shall allow the train to open the specific door in the side of platform.

* + - 1. Vital passenger exchange zone

The LEFT or RIGHT, defined as attributes of VPEZ, PSD or EVAC singularities in track map, means the platform is on the left of right side of the train seen as towards the UP orientation.

[iTC\_CC\_ATP-SwRS-0262]

ATP shall determine whether there are intersection between the range of train locations and the vital passenger exchange zone of the side-A:

In initialization or train delocalization, set the [TrainInterVPEZ\_A](#TrainInterVPEZ_A) as False;

If there are intersection between the range of train locations (from the minimum train tail to the maximum train head) and the vital passenger exchange zone (composing by a pair of SGL\_VPEZ singularities with opposite direction in the train map), and the train door side A correspond to the platform according with [Table 512](#Ref3533565863), set [TrainInterVPEZ\_A](#TrainInterVPEZ_A) as True:

Otherwise, set [TrainInterVPEZ\_A](#TrainInterVPEZ_A) as False.

[iTC\_CC\_ATP-SwRS-0263]

ATP shall determine whether there are intersection between the range of train locations and the vital passenger exchange zone of the side-B:

In initialization or train delocalization, set the [TrainInterVPEZ\_B](#TrainInterVPEZ_B) as False;

If there are intersection between the range of train locations (from the minimum train tail to the maximum train head) and the vital passenger exchange zone (composing by a pair of SGL\_VPEZ singularities with opposite direction in the train map), and the train door side B correspond to the platform according with [Table 512](#Ref3533565865), set [TrainInterVPEZ\_B](#TrainInterVPEZ_B) as True:

Otherwise, set [TrainInterVPEZ\_B](#TrainInterVPEZ_B) as False.

[iTC\_CC\_ATP-SwRS-0560]

[VPEZid\_A](#VPEZid_A) records the ID of the VPEZ, which intersected with the location of train on side A.

If the [TrainInterVPEZ\_A](#TrainInterVPEZ_A), set as the ID of this VPEZ.

Otherwise, set as None.

[iTC\_CC\_ATP-SwRS-0561]

[VPEZid\_B](#VPEZid_B) records the ID of the VPEZ, which intersected with the location of train on side B.

If the [TrainInterVPEZ\_B](#TrainInterVPEZ_B), then set as the ID of this VPEZ.

Otherwise, set as None.

[iTC\_CC\_ATP-SwRS-0264]

[TrainIncludedInVPEZ\_A](#TrainIncludedInVPEZ_A), ATP shall determine whether the whole train location belongs to one vital passenger exchange zone [VPEZid\_A](#VPEZid_A).

In initialization or train delocalization, set the [TrainIncludedInVPEZ\_A](#TrainIncludedInVPEZ_A)as False;

If the range of train locations (from the minimum train tail to the maximum train head) belongs to a VPEZ, and the [TrainInterVPEZ\_A](#TrainInterVPEZ_A) is True, then ATP shall set [TrainIncludedInVPEZ\_A](#TrainIncludedInVPEZ_A) as True;

Otherwise, set [TrainIncludedInVPEZ\_A](#TrainIncludedInVPEZ_A) as False.

[iTC\_CC\_ATP-SwRS-0265]

[TrainIncludedInVPEZ\_B](#TrainIncludedInVPEZ_B), ATP shall determine whether the whole train location belongs to one vital passenger exchange zone [VPEZid\_B](#VPEZid_B).

In initialization or train delocalization, set the [TrainIncludedInVPEZ\_B](#TrainIncludedInVPEZ_B)as False;

If the range of train locations (from the minimum train tail to the maximum train head) belongs to a VPEZ, and the [TrainInterVPEZ\_B](#TrainInterVPEZ_B) is True, then ATP shall set [TrainIncludedInVPEZ\_B](#TrainIncludedInVPEZ_B) as True;

Otherwise, set [TrainIncludedInVPEZ\_B](#TrainIncludedInVPEZ_B) as False.

* + - 1. Safety immobilized

The ATP needs to know whether the train has applied the parking or emergency brake, which used to judge the safe immobilization.

[iTC\_CC\_ATP-SwRS-0576]

[TrainEmergencyBrakeApplied](#TrainEmergencyBrakeApplied) shows that whether the train has applied emergency brake.

If the project is not configured, ATP shall consider the emergency brake has not applied by the train.

Otherwise, if either of the end is in emergency brake, ATP considers the emergency brake has applied.

[iTC\_CC\_ATP-SwRS-0073]

The term [TrainParkingBrakeApplied](#TrainParkingBrakeApplied) stands for that either of the train ends is in parking brake.

[iTC\_CC\_ATP-SwRS-0329]

ATP shall consider that train safely immobilized if:

Train brake has detected safely applied, or train parking brake is detected;

And train is detected at filtered stop.

* 1. + 1. Door opening authorization

[iTC\_CC\_ATP-SwRS-0737]

NoVitalCorrectlyDocked

[iTC\_CC\_ATP-SwRS-0330]

LocalATPenableDoorOpening\_A

ATP shall determine whether train doors on side A opening. The rules are following:

[iTC\_CC\_ATP-SwRS-0331]

LocalATPenableDoorOpening\_B

ATP shall determine whether train doors on side B opening. The rules are following:

[iTC\_CC\_ATP-SwRS-0720]

EnableDoorOpening\_A

[iTC\_CC\_ATP-SwRS-0721]

EnableDoorOpening\_B

5.36.4. Outputs

* 1. F62-Control PSD Opening and Closing Order

According to the intersection condition between train location and PSD area, ATP shall judge to establish or end the communication with the specified interlock devices. After establishment succeeded, ATP should send the PSD order come from CCNV to the interlock through safe communication protocols. ATP can conduct the MAX\_CONNECTED\_PSD\_NB with interlock at the same time.

* + 1. Inputs
    2. Locals
    3. Processing
       1. Communicate with PSD

[iTC\_CC\_ATP-SwRS-0266]

[AlignPSDzone\_A](#AlignPSDzone_A), ATP shall determine whether there are intersection between the range of train locations and the platform screen doors zone of the side-A.

[PSDid\_A](#PSDid_A), the id of the PSD on side-A intersects with train location.

In initialization or train delocalization, set [AlignPSDzone\_A](#AlignPSDzone_A) as False and [PSDid\_A](#PSDid_A) as None.

If there are intersection between the range of train locations (from the minimum train tail to the maximum train head) and the PSD zone (composing by a pair of SGL\_PSD\_ZONE singularities with opposite direction in the train map), and the train door side A correspond to the platform according with [Table 512](#Ref3533565867), set [AlignPSDzone\_A](#AlignPSDzone_A) as True, and records [PSDid\_A](#PSDid_A) as the PSD’s id:

Otherwise, set [AlignPSDzone\_A](#AlignPSDzone_A) as False and [PSDid\_A](#PSDid_A) as None.

[iTC\_CC\_ATP-SwRS-0268]

[AlignPSDzone\_B](#AlignPSDzone_B), ATP shall determine whether there are intersection between the range of train locations and the platform screen doors zone of the side-A.

[PSDid\_B](#PSDid_B), the id of the PSD on side-A intersects with train location.

In initialization or train delocalization, set [AlignPSDzone\_B](#AlignPSDzone_B) as False and [PSDid\_B](#PSDid_B) as None.

If there are intersection between the range of train locations (from the minimum train tail to the maximum train head) and the PSD zone (composing by a pair of SGL\_PSD\_ZONE singularities with opposite direction in the train map), and the train door side B correspond to the platform according with [Table 512](#Ref3533565869), set [AlignPSDzone\_B](#AlignPSDzone_B) as True, and records [PSDid\_B](#PSDid_B) as the PSD’s id:

Otherwise, set [AlignPSDzone\_B](#AlignPSDzone_B) as False and [PSDid\_B](#PSDid_B) as None.

[iTC\_CC\_ATP-SwRS-0136]

[PSDoperation\_A](#PSDoperation_A) and [PSDoperation\_B](#PSDoperation_B) structured as [ST\_PSD\_OPERATION](#bookmark730), used to obtain the PSD controlling order from CCNV.

[iTC\_CC\_ATP-SwRS-0467]

ATP shall determine whether to establish communication with the correlative CI according to request from CCNV:

When there is at least one id of PSDoperation\_A or PSDoperation\_B is not none, ATP shall set *CommunicatedWithPSD* to True:

Otherwise, set *CommunicatedWithPSD* to False.

When the train is running in the station equipped with PSD, ATP needs to communicate with CI and get the PSD status information. According to the configuration data, ATP will process the PSD from CI, and the index is the id of PSD, just as shown in the [ST\_PSD\_TABLE](#bookmark734)

[iTC\_CC\_ATP-SwRS-0111]

In initialization, all PSD DoorClosed of [TableOfPSDPlatform](#TableOfPSDPlatform) set as False, and the valid period is zero. Later on, if receiving correct [CI\_IOstatus](#CI_IOstatus), on the basis of ID number the status of related PSD doorClosed from [TableOfPSDPlatform](#TableOfPSDPlatform) is updated and the DoorStatusValidityTime is set as [ATPsetting](#ATPsetting).PSDstatusValidityTime subtracted the delay of FSFB2 message.

[iTC\_CC\_ATP-SwRS-0112]

If ATP does not receive the [CI\_IOstatus](#CI_IOstatus) from CI, or if this message detected as False, ATP shall update the PSD status of [TableOfPSDPlatform](#TableOfPSDPlatform), i.e. it is necessary to minus DoorStatusValidityTime with one. If the value is less than or equal to zero, the status of DoorClosed is set as False; otherwise the status keeps the same.

* + - 1. PSD status

In CBTC mode, when train is moving, ATP shall monitor the PSD status by the variant from [ZC](#bookmark736); and on the other hand, when train is stopping or just beginning to start, ATP shall use the PSD status adopted from the communication with interlock device. In the block mode, ATP will not consider the PSD status when train is moving (i.e. Will consider the PSD is closed and locked); and when train is stopping or starting, ATP will use the PSD information from interlock devices. Therefore, ATP should consider the updating of [PSDzoneStatus\_A](#PSDzoneStatus_A) and [PSDzoneStatus\_B](#PSDzoneStatus_B) based on the whether ATP is master or not, the information from interlock device, [ZC](#bookmark737) or BM.

[iTC\_CC\_ATP-SwRS-0332]

UsingPSDstatusFromCI

Only when the train fulfilled the following conditions, ATP shall use the PSD status from the CI:

The train location intersects with a PSD zone;

And the train is filtered stopped or just started moving.

[iTC\_CC\_ATP-SwRS-0134]

[MasterCCcore](#MasterCCcore) shows whether the status from CCNV is the main controlled CC.

[iTC\_CC\_ATP-SwRS-0155]

[PSDzoneStatus\_A](#PSDzoneStatus_A)

If the ATP is the master CC, then the A-side PSD state updating rules are as follows:

[iTC\_CC\_ATP-SwRS-0532]

[PSDzoneStatus\_B](#PSDzoneStatus_B)

If the ATP is the master CC, then the B-side PSD state updating rules are as follows:

[iTC\_CC\_ATP-SwRS-0533]

If the ATP is not the master CC, then the A-side PSD state updating rules are as follows:

[iTC\_CC\_ATP-SwRS-0534]

If the ATP is not the master CC, then the B-side PSD state updating rules are as follows:

[iTC\_CC\_ATP-SwRS-0722]

PSDstatusNonVital\_A

[iTC\_CC\_ATP-SwRS-0723]

PSDstatusNonVital\_B

* + - 1. PSD order

For the control order of the PSD, ATP shall generate according to the request of the CCNV and the safety communication protocol defined in the project configuration.

[iTC\_CC\_ATP-SwRS-0333]

PSDmanagerOrder\_A

The rules to generate the PSD manage order on side A shall follow the pseudo-codes. In which if the PSD id from CCNV is not equal to the id in ATP's track map, ATP shall prohibit the PSD opening.

In the above ARDL, the Open\_PSD\_Configuration and Close\_PSD\_Of\_Platform are control words defined in the project configuration.

[iTC\_CC\_ATP-SwRS-0334]

PSDmanagerOrder\_B

The rules to generate the PSD manage order on side B shall follow the pseudo-codes. In which if the PSD id from CCNV is not equal to the id in ATP's track map, ATP shall prohibit the PSD opening.

In the above ARDL, the Open\_PSD\_Configuration and Close\_PSD\_Of\_Platform are control words defined in the project configuration.

[iTC\_CC\_ATP-SwRS-0335]

PSDplatformManagerOpeningOrder

ATP shall determine whether itself opening the PSD in this cycle.

* + - 1. Send PSD command

[iTC\_CC\_ATP-SwRS-0336]

PSDopeningCommand

ATP shall determine whether itself or the redundant ATP opening the PSD in this cycle.

[iTC\_CC\_ATP-SwRS-0444]

When communicating with the CI, if the [PSDmanagerOrder\_A](#PSDmanagerOrder_A) which comes from CCNV was valid, ATP shall set the [CIsetting](#CIsetting)[0] according to the [PSDmanagerOrder\_A](#PSDmanagerOrder_A) and the configuration of the PSD.

[iTC\_CC\_ATP-SwRS-0445]

When communicating with the CI, if the [PSDmanagerOrder\_B](#PSDmanagerOrder_B), which comes from CCNV, was valid, ATP shall set the [CIsetting](#CIsetting)[1] according to the [PSDmanagerOrder\_B](#PSDmanagerOrder_B) and the configuration of the PSD.

* + 1. Outputs
  1. F63-Monitor Status of Doors

This module describes the ATP requirements for the monitor status of doors.

* + 1. Inputs
    2. Locals
    3. Processing
       1. Train doors monitoring

When the train stopped and opened the door in the station, ATP shall keep outputting parking brake.

[iTC\_CC\_ATP-SwRS-0070]

The [AllTrainDoorsClosedAndLocked](#AllTrainDoorsClosedAndLocked) stands for the condition that either side of both train ends collect TDCL, i.e. both side of door is closed and locked.

[iTC\_CC\_ATP-SwRS-0337]

ATP shall not monitor the status of train doors when *InhibitControlTrainDoorsStatus* is selected.

[iTC\_CC\_ATP-SwRS-0804]

NoDangerForTrainDoorsNotClosedAndLocked

[iTC\_CC\_ATP-SwRS-0338]

PbforTrainDoorsNotClosedAndLocked

If the train is aligning in a PSD or intersecting with a vital passage exchange zone, and the RMF or RMR does not selected, ATP shall keep triggering parking brake when the train doors does not closed and locked.

[iTC\_CC\_ATP-SwRS-0339]

EBforPBnotAppliedDueToTrainDoors

If ATP has triggered parking brake for train doors opening, but it does not applied by the rolling stock, ATP shall trigger the emergency brake.

[iTC\_CC\_ATP-SwRS-0340]

NoDangerForDepartureWithoutTDCL

ATP shall determine whether the train is departure without TDCL.

[iTC\_CC\_ATP-SwRS-0749]

EBforDepartureWithoutTDCL

If ATP needs to monitor the status of train doors, ATP shall trigger EB if train determine without TDCL:

[iTC\_CC\_ATP-SwRS-0799]

ATP shall not monitor the status of train doors open without door opening enable if [InhibitProtectionMovingWithoutTDCL](#InhibitProtectionMovingWithoutTDCL) is selected.

[iTC\_CC\_ATP-SwRS-0800]

NoDangerForMovingWithoutTDCL

[iTC\_CC\_ATP-SwRS-0801]

EBforMovingWithoutTDCL

* + - 1. PSD status monitoring

In the non-RM mode, or in block mode with PSD communication, ATP shall guarantee the train cannot move while the PSD has opened or the opening PSD order is sending. Otherwise, ATP shall trigger EB.

[iTC\_CC\_ATP-SwRS-0341]

InhibitControlPSDstatus

The conditions ATP does not control PSD can be configured by project.

[iTC\_CC\_ATP-SwRS-0342]

If the train stopped or just started moving, and the status of all aligned PSD are closed, ATP shall consider the [AllPSDclosedAndLocked](#AllPSDclosedAndLocked) is True.

[iTC\_CC\_ATP-SwRS-0750]

NoDangerforUnexpectedPSDopening

ATP shall determine whether the train starts moving without the aligned PSD closed.

[iTC\_CC\_ATP-SwRS-0343]

EBforUnexpectedPSDopening

If in charge of the PSD control, ATP shall trigger emergency brake when train just started moving but PSD does not closed.

[iTC\_CC\_ATP-SwRS-0803]

NoDangerForPSDnotClosedAndLocked

[iTC\_CC\_ATP-SwRS-0344]

PBforPSDnotClosedAndLocked

If one of the statuses of the aligned PSD does not closed when train stopped, ATP shall trigger parking brake.

[iTC\_CC\_ATP-SwRS-0345]

EBforPBnotAppliedDueToPSD

If ATP has triggered the parking brake for the PSD opening, but it does not applied by the rolling stock, ATP shall trigger the emergency brake.

[iTC\_CC\_ATP-SwRS-0346]

PSDopeningCommandValid

ATP shall monitor the PSD opening command whether valid by the following rules:

If the PSD opening command is True in the current cycle, ATP consider this command is valid;

Or else:, if the PSD opening command become from True to False, ATP shall start to counter the cycles:

If the PSD opening command has become False and has lasted more than [ATPsetting](#ATPsetting).PSDopeningCommandValidityTime, then ATP shall consider the PSD opening command as invalid;

Or else:, ATP still consider the PSD opening command as valid.

Otherwise, ATP consider the PSD opening command as invalid.

[iTC\_CC\_ATP-SwRS-0347]

InhibitPSDopeningSupervisedByATP

[iTC\_CC\_ATP-SwRS-0348]

PBforPSDopenedAndSupervisedByATP

If ATP needs to supervise the PSD opening status, ATP shall trigger parking brake when the PSD opening command is valid.

* + 1. Outputs
  1. F64-Protect Passengers during Emergency Evacuation

This module is for emergency evacuation.

* + 1. Inputs
    2. Locals
    3. Processing
       1. Hold detrainment door closed

For the project equipped with the detrained door in driving cab, when the train stopped or dynamic measurement invalid, if emergency handle is pulling down, the ATP shall trigger EB, unlock the door for passengers getting out.

[iTC\_CC\_ATP-SwRS-0072]

[EmergencyHandleNotPulledEnd1](#EmergencyHandleNotPulledEnd1) stands for the closed emergency door of END\_1. If the train does not allocate with emergency door in the cab, it is certain that the emergency door does not opened.

[iTC\_CC\_ATP-SwRS-0724]

[EmergencyHandleNotPulledEnd2](#EmergencyHandleNotPulledEnd2) stands for the closed emergency door of End2. If the train does not allocate with emergency door in the cab, it is certain that the emergency door does not opened.

[iTC\_CC\_ATP-SwRS-0349]

HoldDoorsClosedTrainEnd1

ATP shall keep hold the train END\_1 door closed when one of the following conditions fulfilled:

Train kinematics is valid and the train does not stop;

Or the emergency handle of END\_1 is not pulled;

[iTC\_CC\_ATP-SwRS-0350]

HoldDoorsClosedTrainEnd2

ATP shall keep hold the train END\_2 door closed when one of the following conditions fulfilled:

Train kinematics is valid and the train does not stop;

Or the emergency handle of END\_2 is not pulled;

[iTC\_CC\_ATP-SwRS-0351]

EBforNotAllTrainEndHoldDoorsClosed

If ATP does not hold the train end door, then trigger emergency brake.

[iTC\_CC\_ATP-SwRS-0738]

EmergencyDetrainDoorLockingEnd1

[iTC\_CC\_ATP-SwRS-0739]

EmergencyDetrainDoorLockingEnd2

* + - 1. Hazardous Evacuation zone

[iTC\_CC\_ATP-SwRS-0273]

ATP shall consider the evacuation is not possible on side-A when there are intersection between the range of train locations and the hazardous evacuation zone of the side-A:

In initialization or not [TrainLocatedOnKnownPath](#TrainLocatedOnKnownPath), set the [EvacuationNotPossible\_A](#EvacuationNotPossible_A) as False;

Or else, if there are intersection between the range of train locations (from the minimum train tail to the maximum train head) and the vital passenger exchange zone (composing by a pair of SGL\_HAZAR\_EVAC\_ZONE singularities with opposite direction in the train map), and the train door side A correspond to the EVAC according with [Table 512](#Ref35335658611), set [EvacuationNotPossible\_A](#EvacuationNotPossible_A) as True.

Otherwise, set [EvacuationNotPossible\_A](#EvacuationNotPossible_A) as False.

[iTC\_CC\_ATP-SwRS-0274]

ATP shall consider the evacuation is not possible on side-B when there are intersection between the range of train locations and the hazardous evacuation zone of the side-B:

In initialization or not [TrainLocatedOnKnownPath](#TrainLocatedOnKnownPath), set the [EvacuationNotPossible\_B](#EvacuationNotPossible_B) as False;

Or else, if there are intersection between the range of train locations (from the minimum train tail to the maximum train head) and the vital passenger exchange zone (composing by a pair of SGL\_HAZAR\_EVAC\_ZONE singularities with opposite direction in the train map), and the train door side B correspond to the EVAC according with [Table 512](#Ref35335658613), set [EvacuationNotPossible\_B](#EvacuationNotPossible_B) as True.

Otherwise, set [EvacuationNotPossible\_B](#EvacuationNotPossible_B) as False.

5.39.3.3. Hold train side doors closed

When the train speed is higher than project configuration data, ATP shall request the train to close and lock on both sides of the door. Only when the speed is lower than the value of the project configuration data, and this side permits for the escape (is determined by restricted escape area in the lines), or there is no end door configured in the cab, ATP shall allow to release the door locking request.

[iTC\_CC\_ATP-SwRS-0352]

The conditions ATP determining the [HoldDoorsClosed\_A](#HoldDoorsClosed_A) show as following ARDL:

[iTC\_CC\_ATP-SwRS-0353]

The conditions ATP determining the [HoldDoorsClosed\_B](#HoldDoorsClosed_B) show as following ARDL :

5.39.3.4 Evacuation requirement

When the train is just leaving from the platform or stops out of the station, if the passenger pulls down the emergency handle, ATP shall trigger EB to stop the train. If the train is running on the interval region, when the passenger pulls down the handle, the ATP shall not output EB.

[iTC\_CC\_ATP-SwRS-0354]

TrainDockedInStation

ATP shall determine whether the train has docked in the station correctly according to conditions of train stopping and doors opening enable.

[iTC\_CC\_ATP-SwRS-0355]

The train is said to be leaving the station:

if since last time train has been detected docked in station ([TrainDockedInStation](#TrainDockedInStation)), the cumulated of the absolute value of [MaximumTrainMotion](#MaximumTrainMotion) is in the range [0, [ATPsetting](#ATPsetting).EvacuationStationAreaLength] and no train kinematic invalidation occurs.

or else, if the train kinematics is invalid, ATP shall set [TrainLeavingStation](#TrainLeavingStation) as False and clear the cumulated distance.

[iTC\_CC\_ATP-SwRS-0071]

EmergencyHandleNotPulledSide shows that the emergency handles is not pulled down.

[iTC\_CC\_ATP-SwRS-0356]

If the train is just leaving the station and the side doors emergency handles are pulled, ATP shall require [EvacuationWhileLeavingStation](#EvacuationWhileLeavingStation).

[iTC\_CC\_ATP-SwRS-0357]

If the train does not stop on the doors opening enable area and the side doors emergency handles pulled, ATP shall require [EvacuationWithTrainStopped](#EvacuationWithTrainStopped).

[iTC\_CC\_ATP-SwRS-0726]

InhibitProtectionEvacuationInDistance

[iTC\_CC\_ATP-SwRS-0727]

InhibitProtectionEvacuationWithStop

[iTC\_CC\_ATP-SwRS-0358]

EBforEvacuationWhileTrainLeavingStation

If the train leaving station evacuation has been required, ATP shall trigger the emergency brake.

[iTC\_CC\_ATP-SwRS-0748]

EBforEvacuationWithTrainStopped

If the train stopped evacuation has been required, ATP shall trigger the emergency brake.

* + 1. Outputs
  1. F7-Generate Output Orders
  2. F71-Outputs to Rolling-stock
     1. Inputs
     2. Locals
     3. Processing
        1. Parking brake

ATP shall control the parking brake of the train. Due to the safety oriented, ATP shall set the output port as restricted status when outputting the parking brake order, and set the port as permissible status when the ATP does not send parking brake.

[iTC\_CC\_ATP-SwRS-0133]

*PBforOperationalRequest* stands for the ZVRD output order from CCNV.

[iTC\_CC\_ATP-SwRS-0359]

[TrainParkingBrakeRequested](#TrainParkingBrakeRequested), determine whether to apply parking brake. This variable shall be True when one of the following conditions met:

Train is in front of a possibly non-exclusive route,

Synthesis of speed constraints on the train implies that it is not allowed to move anymore. Any movement may lead to an hazardous situation,

Train is located on a passenger exchange area with PSD and train doors are not proven closed and locked,

Train is located on a passenger exchange area with PSD and PSD are not proven closed and locked,

There is a possibility of undetectable dangers,

An operational parking brake is requested,

The PSD are opened and are under the supervision of ATP,

The EB has been requested in this cycle.

[iTC\_CC\_ATP-SwRS-0360]

[InhibitParkingBrake](#InhibitParkingBrake)，ATP software do not apply the parking brake.

5.41.3.2. Emergency brake

ATP shall control the emergency brake of the train. Due to the safety oriented, ATP will set the output port as restricted status when outputting the emergency brake order, and set the port as permissive status when the ATP does not send trigger the brake.

[iTC\_CC\_ATP-SwRS-0556]

The [Coreld](#bookmark772) and [SubSystemID](#bookmark773) of the ATP and redundant ATP need to compare for the consistency, which records in [IncompatibleDistantATP](#IncompatibleDistantATP). In initialization, the message from redundant ATP cannot be used. On the other hand, when SubSystemID in the Dataplug read by redundant ATP is the same, but the Coreld is different, both ATP regards as consistency. Otherwise, ATP would trigger emergency brake.

[iTC\_CC\_ATP-SwRS-0271]

[ApproachableSignalOverrun](#ApproachableSignalOverrun), ATP shall determine whether the location of maximum train head overruns an approachable signal with overlap established.

When all of the following conditions fulfilled, ATP considers the train has overrun a restricted signal in this cycle, and set [ApproachableSignalOverrun](#ApproachableSignalOverrun) as True.

not [MotionProtectionInhibition](#MotionProtectionInhibition);

And [ApproachableSignalOverrun](#ApproachableSignalOverrun) was False at the last cycle;

And the moving direction in current cycle is toward on the train front end;

And the maximum location of train front end passes the position of the signal with overlap attribute in this cycle;

And the variants status of the signal is overlap established(refer to [Table 59](#Ref3651281892)).

Otherwise, ATP set [ApproachableSignalOverrun](#ApproachableSignalOverrun) as False.

[iTC\_CC\_ATP-SwRS-0132]

EBforOperationalRequest

ATP shall trigger emergency brake according to CCNV‘s operational emergency brake request.

[iTC\_CC\_ATP-SwRS-0361]

TrainEmergencyBrakeRequested

ATP shall control emergency brake output according following emergency braking requests from control functions:

moral-time control function has detected an hazardous situation (route exclusivity violation);

train speed is no longer compliant with respect of whole speed restriction of guide way;

an approachable speed limit has been over-run (RM speed limit or memorized location speed limit);

an over-speed in reverse direction of travel has been detected;

an emergency evacuation is required for passengers;

train departure with not all doors closed and locked has been detected;

the train starts to move on a PSD zone which status is not "all PSD proven closed and locked";

train has moved although there are potential undetectable dangers;

an operational emergency braking has been requested by CC-Non Vital;

train end doors are not closed and locked;

not all doors closed and locked has been detected on a PSD zone and parking brake is not applied,

not all PSD closed and locked has been detected on a PSD zone and parking brake is not applied;

the approachable signal is overrun;

the VLE-2 safe timer failed;

the information of Dataplug in both ends of cab is inconsistent.

[iTC\_CC\_ATP-SwRS-0362]

InhibitEmergencyBrake

If an emergency braking request ordered by a control function, ATP shall not inhibit emergency brake until train filtered-stop reached.

ATP shall inhibit emergency brake if and only if train detected at filtered stop and there is no emergency braking request from control functions.

5.41.3.3. Vital outputs

For different project, the internal calculating output ports of ATP are configurable, which defined in the project vital settings.

[iTC\_CC\_ATP-SwRS-0546]

CCworkOvertime

ATP shall monitor the CC continuous work time. If the CC is running more than MAX\_RESET\_TIME (the value is far less than MAX\_ATP\_LOOP\_HOUR), the ATP shall set all output to VIOM as restricted status.

[iTC\_CC\_ATP-SwRS-0584]

VIOM1OutNotDisabled，VIOM2OutNotDisabled

Whether CCNV request the channel of VIOM shall be disabled or not.

NOTES：

When some vital outputs are "not disabled" by CCNV, ATP shall set their values as commands calculated by ATP; and if CCNV disabled these outputs channel, ATP shall set their values as restriction. That is to say, CCNV can coerce some output channels as restricted status (to the safe side), but cannot coerce them as permissive (to the hazardous side). This function normally used for the combined test.

[iTC\_CC\_ATP-SwRS-0465]

VIOM1VitalOut，VIOM2VitalOut

[iTC\_CC\_ATP-SwRS-0455]

According to [[REF5]](#bookmark781), the commands ATP sent to VIOM shall attach the current ATPtime, cycle synchronization check word Trace and [Dt](#Dt), safety clock state [SafeTimerFailed](#SafeTimerFailed), [MasterCCcore](#MasterCCcore), the [LatestVIOM1LoopHourVIOM](#LatestVIOM1LoopHourVIOM) and [LatestVIOM2LoopHourVIOM](#LatestVIOM2LoopHourVIOM) needed to feedback to the respective VIOM.

[iTC\_CC\_ATP-SwRS-0752]

OutOfCode

* + 1. Outputs
  1. F72-Outputs to ZC

ATP shall send the Location Report to [ZC](#bookmark783) as far as possible. After ATP powered up and the communication with LC had fulfilled, and get the current [ZC](#bookmark784) id, it shall communicate with [ZC](#bookmark785). When the train passes across two [ZC](#bookmark786), ATP should also send the Location Report to both [ZC](#bookmark787). The information of Location Report includes train recognition information, ATP status, and train location, so that [ZC](#bookmark788) can calculate the AP scope, properties and EOA of the train.

* + 1. Inputs
    2. Locals
    3. Processing
       1. Location Report creation

[iTC\_CC\_ATP-SwRS-0401]

TrainHeadOrientation

ATP shall send the orientation of minimum location of current active cab id to the [ZC](#bookmark794). For the rule can refer to SwRS-0403.

[iTC\_CC\_ATP-SwRS-0403]

TrainHeadMinLocation

ATP shall send the minimum head location of the active cab to the [ZC](#bookmark796), including the block id and its abscissa. According to [[REF5]](#bookmark797), the unit of the abscissa in Location Report is 0.5 meter, so the ATP needs to convert its internal unit to match that. The conversion shall be safety-oriented, which means the envelope of the train location tend to be "stretched" to the both ends. The rules of conversion are as following ARDL:

In above ARDL, the ABSCISSA\_TO\_HALF\_METER means the coefficient of unit conversion.

[iTC\_CC\_ATP-SwRS-0404]

TrainHeadCoupledStatus

ATP shall send the coupled status of the active train cab to the [ZC](#bookmark805).

[iTC\_CC\_ATP-SwRS-0405]

TrainTailCabId

ATP shall send the current inactive cab id to the [ZC](#bookmark806).

[iTC\_CC\_ATP-SwRS-0406]

TrainTailOrientation

ATP shall send the orientation of the minimum location of inactive cab id to the [ZC](#bookmark807). For the rule can refer to SwRS-0408.

[iTC\_CC\_ATP-SwRS-0408]

TrainTailMinLocation

ATP shall send the minimum tail location of the active cab to the [ZC](#bookmark809), including the block id and its abscissa. According to [[REF5]](#bookmark810), the unit of the abscissa in Location Report is 0.5 meter, so the ATP needs to convert its internal unit to match that. The convertion shall be safety-oriented, which means the envelope of the train location tend to be "stretched" to the both ends. The rules of convertion are as following ARDL:

In above ARDL, the ABSCISSA\_TO\_HALF\_METER means the coefficient of unit convertion.

[iTC\_CC\_ATP-SwRS-0409]

TrainTailCoupledStatus

ATP shall send the coupled status of the inactive train cab to the [ZC](#bookmark818).

[iTC\_CC\_ATP-SwRS-0410]

LocationError

ATP shall send the location error to the [ZC](#bookmark820). According to [[REF5]](#bookmark821), the unit of the location error in Location Report is 0.5 meter, so the ATP needs to convert its internal unit to match that. The convertion shall be safety-oriented, which means the location error tend to be "overestimated".

[iTC\_CC\_ATP-SwRS-0068]

According to the status of [RouteSetNotNeededSendable](#RouteSetNotNeededSendable), ATP can judge whether it is necessary to send RSNN information.

[iTC\_CC\_ATP-SwRS-0135]

NonVitalRouteSetNotNeeded

Whether the CCNV request route set note needed.

[iTC\_CC\_ATP-SwRS-0414]

TrainRouteSetNotNeeded

ATP shall send the route set not needed information to [ZC](#bookmark824).

[iTC\_CC\_ATP-SwRS-0415]

TrainCorrectDocking

ATP shall send the docking correction information to the [ZC](#bookmark825).

[iTC\_CC\_ATP-SwRS-0416]

LocReportSpeed

ATP shall send the maximum train speed to the [ZC](#bookmark826). According to [[REF5]](#bookmark827), the unit of the speed in Location Report is KPH, so the ATP needs to convert its internal unit to match that. The convertion shall be safety-oriented, which means the speed tend to be "overestimated".

[iTC\_CC\_ATP-SwRS-0417]

TrainMonitoringMode

ATP shall send the current monitoring mode to the [ZC](#bookmark829).

[iTC\_CC\_ATP-SwRS-0599]

SignalOverrideSendable

[iTC\_CC\_ATP-SwRS-0418]

SignalsOverride

ATP shall send the signal override information to the [ZC](#bookmark831).

[iTC\_CC\_ATP-SwRS-0598]

ATCcontrolledTrain

5.42.3.2. Track side time

[iTC\_CC\_ATP-SwRS-0728]

DateSynchronizationReceived

[iTC\_CC\_ATP-SwRS-0094]

[DateSynchronisationReportAvailable](#DateSynchronisationReportAvailable) used to judge whether the LC synchronization message could use or not. When the below conditions fulfilled, [DateSynchronisationReportAvailable](#DateSynchronisationReportAvailable) shall set as True. Otherwise, it set as False.

If ATP Receives the [DateSynchronizationReport](#DateSynchronizationReport) message from LC, and the checksum is correct.

And If the received message is LC to respond the local CC, ATP shall qualify with below conditions:

The current ATP time is more than ccLoopHour in the message

The different value between the current ATP time and the ccLoopHour is less than the LC message time in the previous ATP.

Or else:, If the received message is LC to respond the remote CC, and it should be qualified with below conditions:

The minimum time in remote ATP maintained by current ATP is more than the ccLoopHour

The different value between maximum time in remote ATP maintained by current ATP and the ccLoopHour is less than the LC message time in the previous ATP.

[iTC\_CC\_ATP-SwRS-0095]

[LastSynchronisationReportAge](#LastSynchronisationReportAge) records the age from previous LC message to current cycle. It used to compare whether the new received message is more updating than last recorded one, to prevent the inverse transition in the network communication.

[iTC\_CC\_ATP-SwRS-0097]

[LCsynchronisationLoopHourValid](#LCsynchronisationLoopHourValid) will check whether the LC message is still valid.

[iTC\_CC\_ATP-SwRS-0098]

ATP shall estimate a time called [GroundTimeReference](#GroundTimeReference) for trackside equipment ([ZC](#bookmark840) /LC). From power-up, [GroundTimeReference](#GroundTimeReference) shall consider invalid. The rules for time information are as follows:

When LC message invalid, it is set as default value.

When ATP receives an available LC message, it updates as the time of message.

Otherwise, it updates the value according to the on board and track side cycle.

NOTES:

In ATP shall not use [GroundTimeReference](#GroundTimeReference) during the calculation of the internal functions, but only send the info to [ZC](#bookmark841).

5.42.3.3. Location Report sendable

[iTC\_CC\_ATP-SwRS-0437]

EnableSendLocReport

Before received the first valid synchronization from a LC, the corresponding LC vital time shall consider not significant, and the ATP shall prevent to send Location Report message to the ZC(s) associated to the corresponding LC.

[iTC\_CC\_ATP-SwRS-0438]

TimeElapseBetweenTwoLocReport，

ATP shall calculate when to send the Location Report as the following ARDL:

[iTC\_CC\_ATP-SwRS-0439]

ZCidUnderTrainTail，

ATP shall record the [ZC](#bookmark843) area id where the train tail located. If the train is not localized, ATP shall use the current receiving beacon where located, or the last known [ZC](#bookmark844) area id.

[iTC\_CC\_ATP-SwRS-0440]

ZCidUnderTrainHead，

ATP shall record the [ZC](#bookmark849) area id where the train head located. If the train is not localized, ATP shall use the current receiving beacon where located, or the last known [ZC](#bookmark850) area id.

[iTC\_CC\_ATP-SwRS-0441]

SendLocReportOnZCunderTrainTail，

When the time elapsed from last report is equal to the Location Report sending cycle, and the train tail located [ZC](#bookmark855) is valid and not as same as the train head [ZC](#bookmark856), and the [ZC](#bookmark857) train tail located authorized by the LC, ATP shall send the Location Report to this [ZC](#bookmark858) where train tail located.

[iTC\_CC\_ATP-SwRS-0442]

SendLocReportOnZCunderTrainHead，

When the time elapsed from last report is equal to the Location Report sending cycle, and the train head located [ZC](#bookmark860) is valid, and the [ZC](#bookmark861) train head located authorized by the LC, ATP shall send the Location Report to this [ZC](#bookmark862) where train head located.

* + 1. Outputs
  1. F73-Outputs to CI Radio
     1. Inputs
     2. Locals
     3. Processing
        1. Receive CC variant request

[iTC\_CC\_ATP-SwRS-0729]

CCvariantRequestMsgReceived

[iTC\_CC\_ATP-SwRS-0730]

ReceivedCBIloopHour

5.43.3.2. Send CC variant report

[iTC\_CC\_ATP-SwRS-0731]

CCvariants，

ATP shall check the following conditions when sending overlap release to CBI:

Train front location is in overlap release zone,

and the other ATP's overlap timer has expired,

and ATP received variant request from CBI in this zone.

* + 1. Outputs
  1. F8-Platform Relative & Assist Functions
  2. F81-Initialize ATP software

This function used to initialize the ATP software.

* + 1. Inputs
    2. Processing
       1. Read data plug

[iTC\_CC\_ATP-SwRS-0511]

In Initialization, ATP reads the [CCdataPlugInfo](#CCdataPlugInfo) with VCP coded from CC data plug, and generates [DataPlugContent](#DataPlugContent) with the structure shown as Table 44。

[iTC\_CC\_ATP-SwRS-0037]

Through [DataPlugContent](#DataPlugContent).VLECpuId, ATP recognizes whether it is located in CPU1 or CPU2, and establish the communication between each other.

If the CPU id is wrong, ATP shall stop and wait for reboot manually by the operator.

For CPU1, ATP shall control the LED to show WAITING\_CPU when the communication establishing.

For CPU1, ATP shall control the LED to show ERR\_CPU\_ID when the cpuId is neither CPU1 nor CPU2.

NOTES:

Due to the limitation of hardware, only the software running on CPU1 can control the display of LED in the front panel of VLE-2 board.

5.45.2.2. CCNV initialization

[iTC\_CC\_ATP-SwRS-0056]

In initialization, ATP needs to send some part of contents as shown in [Table 47](#Ref2882304154) from CC data plug to CCNV by dual-ports RAM after getting contact with CCNV. If ATP cannot establish the communication with CCNV, it will keep waiting and show in the LED as WAITING \_CCNV, until the communication is built or the operator reboots VLE-2 board.

5.45.2.3. DLU initialization

[iTC\_CC\_ATP-SwRS-0512]

In initialization, ATP needs to send the [DataPlugContent](#DataPlugContent). DLUIpBlue and DLUIpRed from CC data plug to DLU through RAM.

5.45.2.4. VPB initialization

[iTC\_CC\_ATP-SwRS-0039]

The hardware needs to be initialized, e.g. VPB board. If the initialization of hardware succeeds, the system will continue. Otherwise, ATP on CPU1 shall show the message in the LED as ERR\_INIT\_VPB, and prohibit software executing and waiting for the system reboot by the operator.

[iTC\_CC\_ATP-SwRS-0591]

In initialization, ATP in two CPU modules shall compare the values from VPB registers. If these initial values are equal, the ATP will continue; Otherwise, ATP shall show the message in LED as UNCONST\_VPB, and prohibit software executing.

* + 1. Outputs
  1. F82-Manage Vital Time
     1. Inputs
     2. Locals
     3. Processing

Both CPU modules of the VLE-2 board have a safe clock respectively. When the ATP software response the fixed-time interrupts, it shall get the number of pulses and determine its correctness. Furthermore, the ATP software on two CPU shall synchronize their length of cycle, and ensure the difference is not exceeding the time of one interrupt. From the safety oriented, the ATP cycle time length only can be overestimated but cannot be underestimated.

5.46.3.1. Immediate Task Count

[iTC\_CC\_ATP-SwRS-0045]

[VLEimpulseNb](#VLEimpulseNb)[ATP\_INTERRUPT\_NB] array stores the safe clock impulse number for every interrupt triggered. ATP shall obtain the impulse number through [VLE\_GetSafeTime](#bookmark897), and stores into the array.

[iTC\_CC\_ATP-SwRS-0756]

*ImmediateCounter*, as the interrupt counter, ATP shall accumulate 1 after each interrupt triggered.

[iTC\_CC\_ATP-SwRS-0046]

When the fixed-time interrupt triggered, ATP shall update the [ImmediateNb](#ImmediateNb)，[CycleSynchronized](#CycleSynchronized), [Trace](#Trace), [Dt](#Dt) and [CycleBiasNb](#CycleBiasNb).

If it is the first interrupt after powered up, ATP shall set [ImmediateNb](#ImmediateNb) as zero，and initialize the [Trace](#Trace), [Dt](#Dt) (the dynamic time of VCP), and the middle variables m based on CPU.

Or else:, If the [ImmediateNb](#ImmediateNb) is zero, but the [VitalTime](#VitalTime) has not changed comparing to the previous interrupt, then:

Or else, If [ImmediateNb](#ImmediateNb) is zero, and the [VitalTime](#VitalTime) has changed comparing to the previous interrupt, then:

Otherwise, set:

and if the [ImmediateNb](#ImmediateNb) is greater than (ATP\_INTERRUPT\_NB-1), then:

NOTES**：**

The numbers of data is classified by the VCP tool beforehand which includes: CPU1\_TRACE\_0, CPU2\_TRACE\_0, CPU1\_TRACE\_N, CPU2\_TRACE\_N, CPU1\_DT\_INIT, CPU2\_DT\_INIT, and Bi[ATP\_INTERRUPT\_NB]. and PDoperation() represents the PD calculation without [Dt](#Dt)；PDoperationDt() stands for the PD calculation with [Dt](#Dt) ; InversePDoperation() is regarded as PD inverse calculation.

5.46.3.2. Deferred Task Control

[iTC\_CC\_ATP-SwRS-0535]

The [SafeTimerFailed](#SafeTimerFailed) defines whether the fix-time interrupt for hardware is correct or not. ATP shall check whether the safe clock impulse number with the adjacent interrupt is within the error range [MIN\_TIMER\_IMPULSE\_NB, MAX\_TIMER\_IMPULSE\_NB].

Once the [SafeTimerFailed](#SafeTimerFailed) was True, ATP shall keep it as True unless the system is rebooted.

Or else:, if the impulse number is within the above-mentioned range, ATP shall set [SafeTimerFailed](#SafeTimerFailed) as False

Otherwise, it will set [SafeTimerFailed](#SafeTimerFailed) as True.

[iTC\_CC\_ATP-SwRS-0047]

The [VitalTime](#VitalTime) stand for the current cycle of ATP deferred task. After power up, it starts from zero and increase one each cycle. When all the work is executed in the main task, ATP detects whether the interrupt task is over, i.e. the difference between [ImmediateCounter](#ImmediateCounter) and [LockedImmediateCounter](#LockedImmediateCounter) is equal to or larger than (ATP\_INTERRUPT\_NB -1).

If it is so, it shows that the main task in this cycle finishes. Then ATP shall:

send the [VitalTime](#VitalTime) to the other CPU,

and send [Trace](#Trace) and [Dt](#Dt) to VIOM to check,

and at the end of this cycle, set

Otherwise, keep waiting.

[iTC\_CC\_ATP-SwRS-0048]

ATP executes the new cycle DeferredTask when below conditions fulfilled:

[CycleSynchronized](#CycleSynchronized) is True,

and the [TOC\_VitalTime](#bookmark899) get from the other CPU is equal to [VitalTime](#VitalTime), representing that the other CPU finished to execute the task in the previous cycle.

If above condition does not fulfill, ATP shall prohibit to execute, and CPU1 shows the message ERR\_SYNCH in the LED of VLE board.

NOTES:

As shown in [Figure 523](#Ref3450646162), this is the demonstration of the process for cycle synchronization between two CPU modules. In cycle N, if the clock of CPU1 was faster than CPU2, then the interrupt counter of CPU1 counts to ATP\_INTERRUPT\_NB and sends the CPU1\_VitalTime(N) information to CPU2. However, at this moment the interrupt counter of CPU2 still do not achieve the new cycle, so both of the CPU module cannot enter into cycle(N+1). Only when the CPU2 has get all interrupt finished signal, then sends the VitalTime(N) to CPU1, and the both CPUs are enter the cycle(N+1) together. During this process, CPU1 has ignored one interrupt period to wait the CPU2.

This kind of "waiting" synchronous way will avoid the danger that the clock is getting faster and faster. That is to say, we may use the measured movement in the period (ATP\_CYCLE\_TIME + ATP\_INTERRUPT\_TIME) divide a fixed ATP\_CYCLE\_TIME, then we can get speed which is higher than the actual value, meaning that we over-evaluate the train speed so as to ensure the safety.

The cycle error between CPU1 and CPU2 cannot exceed one ATP\_INTERRUPT\_TIME. If it exceeded this period, the [CycleSynchronized](#CycleSynchronized) shall be set as False, and the ATP will not execute continuously so that the VIOM will cut off the output to the train.

[iTC\_CC\_ATP-SwRS-0589]

5.46.3.3. RTC time

[iTC\_CC\_ATP-SwRS-0446]

ATP software shall maintain the RTC time for non-vital functions.

In initialization, ATP software get RTC time from VLE-2 board;

And then, ATP software updates the RTC time every second;

And if the difference between RTC time ATP used and the NTP time CCNV sent is greater than MAX\_NTP\_TIME\_ERROR, ATP shall reset the RTC time as NTP time.

The Time.Update() means ATP software shall update the RTC time every second.

* + 1. Outputs
  1. F83-Manage Synchronization between two CPUs
     1. Inputs
     2. Processing

Refer to [Figure 524](#Ref3450646421), the two set of ATP software in the same VLE-2 board will calculate the output results to VIOM, redundant ATP, [ZC](#bookmark914) or LC, and those data will exchanged by DPRAM. Then the output will merge into one pack of data and send to CCNV.

[iTC\_CC\_ATP-SwRS-0050]

In every cycle, ATP combines the vital outputs generated by itself and from the other CPU into one frame and sends it to CCNV, who will transmit the frame to VIOM. ATP will send [VIOM1VitalOut](#VIOM1VitalOut) and [VIOM2VitalOut](#VIOM2VitalOut) to the other CPU through dual-ports RAM, and receive the [TOC\_VIOM1VitalOut](#bookmark917) and [TOC\_VIOM2VitalOut](#bookmark918) from the other CPU.

[iTC\_CC\_ATP-SwRS-0051]

Based on [ST\_VIOM\_OUT](#bookmark923) structure, ATP combines [VIOM1VitalOut](#VIOM1VitalOut) and [TOC\_VIOM1VitalOut](#bookmark924) as [IdenticalVIOM1Out](#IdenticalVIOM1Out), while based on the same structure [ST\_VIOM\_OUT](#bookmark925), ATP combines [VIOM2VitalOut](#VIOM2VitalOut) and [TOC\_VIOM2VitalOut](#bookmark926) as [IdenticalVIOM2Out](#IdenticalVIOM2Out).

[iTC\_CC\_ATP-SwRS-0579]

[IdenticalLocReport](#IdenticalLocReport), the location report after merging two CPU’s results. When the two CPUs complete [LocReport](#LocReport) calculation, need to send each other through the dual-port RAM; and receive the [TOC\_LocReport](#bookmark929) from the other. ATP shall combine the two reports as an [IdenticalLocReport](#IdenticalLocReport), according to the following rules:

The variables without vital checksums in [ST\_LOC\_REPORT](#bookmark930), shall use the values calculated by itself.

For the *VitalChecksum\_1*, ATP shall calculate using the values from CPU1.

For the *VitalChecksum\_2*, ATP shall calculate using the values from CPU2.

NOTES:

Normally, the calculated results between CPU1 and CPU2 should be the same – if not, the receiver will get the wrong checksum calculation and thus reject the message. However, due to the independent interrupts of two CPUs, the moment ATP software reading the odometer information from VPB’s registers is not simultaneous, which lead to unavoidable biases in movement or speed calculation by the two ATP software. Therefore, ATP need to synchronize the VPB inputs (SwRS-0590) and then calculate with [IdenticalLockedOdometer](#IdenticalLockedOdometer).

[iTC\_CC\_ATP-SwRS-0580]

[IdenticalVersionReport](#IdenticalVersionReport), the version report after merging two CPU’s results. When the two CPUs complete [VersionFromCCreport](#VersionFromCCreport) calculation, need to send each other through the dual-port RAM; and receive the [TOC\_VersionReport](#bookmark933) from the other. ATP shall combine the two reports as an [IdenticalVersionReport](#IdenticalVersionReport), according to the following rules:

The variables without vital checksums in [ST\_VERSION\_REPORT](#bookmark934), shall use the values calculated by itself.

For the *VitalChecksum\_1*, ATP shall calculate using the values from CPU1.

For the *VitalChecksum\_2*, ATP shall calculate using the values from CPU2.

[iTC\_CC\_ATP-SwRS-0581]

[IdenticalCCsyncReport](#IdenticalCCsyncReport), the redundant ATP report after merging two CPU’s results. When the two CPUs complete [CCsynchroReport](#CCsynchroReport) calculation, need to send each other through the dual-port RAM; and receive the [TOC\_CCsyncReport](#bookmark937) from the other. ATP shall combine the two reports as an [IdenticalCCsyncReport](#IdenticalCCsyncReport), according to the following rules:

The variables without vital checksums in [ST\_SYNCHRO\_REPORT](#bookmark938), shall use the values calculated by itself.

For the *VitalChecksum\_1*, ATP shall calculate using the values from CPU1.

For the *VitalChecksum\_2*, ATP shall calculate using the values from CPU2.

[iTC\_CC\_ATP-SwRS-0588]

[IdenticalCIsetting](#IdenticalCIsetting), the PSD control message after merging two CPU’s results. When the two CPUs complete [CIsetting](#CIsetting) calculation, need to send each other through the dual-port RAM; and receive the [TOC\_CIsetting](#bookmark941) from the other. ATP shall combine the two reports as an [IdenticalCIsetting](#IdenticalCIsetting), according to the following rules:

The variables in [ST\_CI\_SETTING](#bookmark942), shall use the values calculated by itself.

For the checkword 1 in FSFB2 protocol, ATP shall calculate using the values from CPU1.

For the checkword 2 in FSFB2 protocol, ATP shall calculate using the values from CPU2.

[iTC\_CC\_ATP-SwRS-0746]

[IdenticalCBIvariantRequest](#IdenticalCBIvariantRequest), the CBI variants request after merging two CPU’s results. When the two CPUs complete [CBIvariantRequest](#CBIvariantRequest)calculation, need to send each other through the dual-port RAM; and receive the [TOC\_CCvariantReport](#bookmark944) from the other. ATP shall combine the two reports as an [IdenticalCBIvariantRequest](#IdenticalCBIvariantRequest), according to the following rules:

The variables without vital checksums, shall use the values calculated by itself.

For the *VitalChecksum\_1*, ATP shall calculate using the values from CPU1.

For the *VitalChecksum\_2*, ATP shall calculate using the values from CPU2.

[iTC\_CC\_ATP-SwRS-0747]

[IdenticalCCvariantReport](#IdenticalCCvariantReport), the CC overlap releasable report after merging two CPU’s results. When the two CPUs complete [CCvariantReport](#CCvariantReport)calculation, need to send each other through the dual-port RAM; and receive the [TOC\_CCvariantReport](#bookmark946) from the other. ATP shall combine the two reports as an [IdenticalCCvariantReport](#IdenticalCCvariantReport), according to the following rules:

The variables without vital checksums, shall use the values calculated by itself.

For the *VitalChecksum\_1*, ATP shall calculate using the values from CPU1.

For the *VitalChecksum\_2*, ATP shall calculate using the values from CPU2.

* + 1. Outputs

1. Appendices
   1. Project Configuration of ATPsetting

The project configurable vital settings are shown in [Table 61](#Ref2831100004).

* 1. Carborne Controller Constants

[Table 62](#Ref2801762411) shows the constants used by ATP software, which are commonly used for all iTC projects.

* 1. Logical Types Definition

This part defines variables and structures types used in the description of ATP software requirements.

6.3.1. Fundamental types

6.3.2. Enumerated types

6.3.3. Aggregated types

6.4. Offline generated codes

6.5. Message

Message.Exists, ATP shall determine whether receive a correct message from CCNV.

Message.VitalChecksumValid，When ATP received a safety message, shall check the CRC and calculate the check words of the message. If either the CRC or the vital checkword failed, ATP shall discard the message and return False; Otherwise, ATP shall consider the message is correct and return True.

Message. ReplyLocalCC，Message. ReplyDistantCC，Message. ModularSub，Message.IsMoreRecent

NOTES**：**[ATP\_MESSAGE\_MAX\_DELAY](#bookmark1098) is a constant for the modular subtraction, which shall set as big enough and much less than the MAX\_ATP\_LOOP\_HOUR. There is an assumption that ATP do not need to consider the difference between two loop hour is more than or equal to [ATP\_MESSAGE\_MAX\_DELAY](#bookmark1099), when calculate the modular minus.

Message.Received，Message.Available，Message.LastAge，Message. Valid

* 1. Track Map
     1. Block related

TrackMap.Blocks，TrackMap.BlockOrigin，TrackMap.NextBlock

* + 1. Singularity related

TrackMap.AllSingsBtwTwoLocs，TrackMap.AllSingsInZone，TrackMap. AllSwitchesInZone，TrackMap.ExistSingBtwTwoLocs，TrackMap.ExistSingularityInZone，TrackMap.ExistSingularityInReverseZone，

TrackMap.ExistSwitchBtwTwoLocs， TrackMap.ExistThePole，TrackMap.ExistZoneLocationIncluded

6.6.3.Beacon related

TrackMap.AreNeighbouredBeacons，TrackMap.Beacons，TrackMap.BeaconBelongsToCalibrationCouple，TrackMap.BmBeaconDirection，TrackMap.CalibrationCoupleMaxDistance，

TrackMap.CalibrationCoupleMinDistance，

TrackMap.IsBmBeacon，TrackMap.IsCalibrationValidationBeacon，TrackMap.OrientationOfNeighbouredBeacons，

6.6.4.CBI related

TrackMap.CbiId，TrackMap.NumberOfVariants，

6.6.5.ZC related

TrackMap.ZC，TrackMap.ZCId

6.6.6.Location relationship

TrackMap.CalculateZoneBorder，TrackMap.DistanceBtwTwoLocs，TrackMap.IntersectionOfTwoZones，TrackMap.LocationBtwTwoLocs，TrackMap.LocationInZone，TrackMap.LocationUpdateExt2，TrackMap.OppositeOrientation，TrackMap.ReachableBetweenTwoLocations，

* 1. Assessment on Compliance with EN50128

Compliance of software development with techniques and measures recommended by CENELEC standard EN 50128(2001) is established through correspondence with tables of EN50128(2001) Annex A.

The names of the tables, the references and the conventions are those of the CENELEC standard. A selection of techniques and measure for software development in CC subsystem is given as the following choices:

√’ – means this technique or measure has been applied.

‘-’ – means this technique or measure has not been chosen.

‘NA’ – means this technique or measure is not applicable.