ARINC Standard – Errata Report

1. Document Title

(Insert the number, supplement level, date of publication, and title of the document with the error)

ARINC SPECIFICATION 653P1-3
AVIONICS APPLICATION SOFTWARE STANDARD INTERFACE PART 1 – REQUIRED SERVICES
PUBLISHED: November 15, 2010

Errata 1

2. Reference

Page Number: 25 Section Number: 2.3.2.2.1.3 State Transitions Date of Submission: 11/11/2014

3. Error

(Reproduce the material in error, as it appears in the standard.)

(9c) Waiting - Waiting
When the process is suspended while the partition is in Initialization phase.

4. Recommended Correction

(Reproduce the correction as it would appear in the corrected version of the material.)

(9c) Waiting - Waiting
When the process is suspended or resumed while the partition is in Initialization phase.

5. Reason for Correction(Optional)

(State why the correction is necessary.)

In the COLD/WARM START mode, a *suspended* process can also be *resumed* and its state also transits from *Waiting* to *Waiting.* According to the service specification of the “RESUME” (3.3.2.7 RESUME in page 60), we could see that the “RESUME” service allows resuming a suspended process in the COLD or WARM START mode.

Errata 2

2. Reference

Page Number: 26 Section Number: 2.3.2.2.1.3 State Transitions Date of Submission: 11/11/2014

3. Error

(Reproduce the material in error, as it appears in the standard.)

(11b) Dormant - Waiting
When a periodic process is started (either with a delay or not) and the partition is in NORMAL mode.

4. Recommended Correction

(Reproduce the correction as it would appear in the corrected version of the material.)

(11b) Dormant - Waiting
When a periodic process is started (either with a delay or not) or an aperiodic process is delayed started (when the delay time > 0), and the partition is in NORMAL mode.

5. Reason for Correction(Optional)

(State why the correction is necessary.)

In the NORMAL mode, an aperiodic process can be *delayed started* and its state also transits from *Dormant* to *Waiting.* According to the service specification of the "DELAYED_START" (3.3.2.11 DELAYED_START in page 63), we could see that an aperiodic process is in *Waiting* state after being delayed started.

Errata 3

2. Reference

Page Number: 24 Section Number: 2.3.2.2.1.3 State Transitions Date of Submission: 11/11/2014

3. Error

(Reproduce the material in error, as it appears in the standard.)

(1) Dormant - Ready
When the process is started by another process while the partition is in NORMAL mode.

4. Recommended Correction

(Reproduce the correction as it would appear in the corrected version of the material.)

(1) Dormant - Ready
When the aperiodic process is started or delayed started (when the delay time = 0) while the partition is in NORMAL mode.

5. Reason for Correction(Optional)

(State why the correction is necessary.)

It is obvious that the periodic process will transit to the *Waiting* state after being started (either with a delay or not). Second, according to the service specification of the "DELAYED_START" (3.3.2.11 DELAYED_START in page 63), we could see that an aperiodic process is in *Ready* state after being delayed started (when delay time =0).

Errata 4

2. Reference

Page Number: 53 Section Number: 3.2.2.2 SET_PARTITION_MODE Date of Submission: 11/11/2014

3. Error

(Reproduce the material in error, as it appears in the standard.)

```
procedure SET_PARTITION_MODE
      (OPERATING_MODE : in OPERATING_MODE_TYPE;
      RETURN_CODE : out RETURN_CODE_TYPE) is
error
      when (OPERATING_MODE does not represent an existing mode) =>
            RETURN_CODE := INVALID_PARAM;
      when (OPERATING_MODE is NORMAL and current mode is NORMAL) =>
            RETURN_CODE := NO_ACTION;
      when (OPERATING_MODE is WARM_START and current mode is COLD_START) =>
            RETURN_CODE := INVALID_MODE;
```

4. Recommended Correction

(Reproduce the correction as it would appear in the corrected version of the material.)

procedure SET_PARTITION_MODE

```
    (OPERATING_MODE : in OPERATING_MODE_TYPE;
    RETURN_CODE : out RETURN_CODE_TYPE) is
error
    when (OPERATING_MODE does not represent an existing mode) =>
        RETURN_CODE := INVALID_PARAM;
    when (OPERATING_MODE is NORMAL and current mode is NORMAL) =>
        RETURN_CODE := NO_ACTION;
    when (OPERATING_MODE is NORMAL and there is not process in current parttion) =>
        RETURN_CODE := INVALID_MODE;
    when (OPERATING_MODE is WARM_START and current mode is COLD_START) =>
        RETURN_CODE := INVALID_MODE;
```

5. Reason for Correction(Optional)

(State why the correction is necessary.)

Beside the main process, a partition must have one or more ARINC processes. Because the main process is the default process of the application, and is the only process eligible for scheduling during the COLD START and WARM START partition operating modes. Therefore, a partition cannot be set into NORMAL mode, if ARINC processes have not been created in this partition.

Errata 5

2. Reference

Page Number: 60 Section Number: 3.3.2.7 RESUME Date of Submission: 11/11/2014

3. Error

(Reproduce the material in error, as it appears in the standard.)

```
if (the specified process is not waiting on a process queue or TIMED_WAIT time delay) then
    set the specified process state to READY;
    if (preemption is enabled) then
        ask for process scheduling;
        -- The current process may be preemptedby the resumed process
    end if;
end if;
```

4. Recommended Correction

(Reproduce the correction as it would appear in the corrected version of the material.)

```
if (the specified process is not waiting on a resource) then
```

```
        set the specified process state to READY;
        if (preemption is enabled) then
            ask for process scheduling;
            -- The current process may be preemptedby the resumed process
        end if;
end if;
```

5. Reason for Correction(Optional)

(State why the correction is necessary.)

An aperiodic process that has been delayed started was suspended, it was in *Waiting* state. Then if we resume this process, it should retain in the *Waiting* state if the delay time has not reached.

Errata 6

2. Reference

Page Number: 78 Section Number: 3.6.2.2.2 SEND_QUEUING_MESSAGE Date of Submission: 11/11/2014

3. Error

(Reproduce the material in error, as it appears in the standard.)

```
if (expiration of the time-out) then
        RETURN_CODE := TIMED_OUT;
else
        -- There is sufficient space in the port's message queue to insert the
          -- message represented by MESSAGE_ADDR and LENGTH in the port's message queue;
        if (TIME_OUT is not infinite) then
            stop the time counter;
        end if;
        RETURN_CODE := NO_ERROR;
end if;
```

4. Recommended Correction

(Reproduce the correction as it would appear in the corrected version of the material.)

```
if (expiration of the time-out) then
        RETURN_CODE := TIMED_OUT;
else
```

-- There is sufficient space in the port's message queue to insert the
-- message represented by MESSAGE_ADDR and LENGTH in the port's message queue;
if (TIME_OUT is not infinite) then
stop the time counter;
end if;
insert the message represented by MESSAGE_ADDR and LENGTH in the FIFO message queue of the specified port;
RETURN_CODE := NO_ERROR;
end if;

5. Reason for Correction(Optional)

(State why the correction is necessary.)

when the time-out does not expire and the space becomes free, the sent message should be inserted into the message queue.

Errata 7

2. Reference

Page Number: 86-87 Section Number: 3.7.2.1.3 RECEIVE_BUFFER Date of Submission: 11/11/2014

3. Error

(Reproduce the material in error, as it appears in the standard.)

normal
if (message buffer is not empty) then
copy the first message of the specified buffer message queue to the location represented by MESSAGE_ADDR;
LENGTH := Length of the copied message;

4. Recommended Correction

(Reproduce the correction as it would appear in the corrected version of the material.)

normal
if (message buffer is not empty) then
copy the first message of the specified buffer message queue to the location represented by MESSAGE_ADDR;
remove the message from the FIFO message queue of the specified buffer
LENGTH := Length of the copied message;

5. Reason for Correction(Optional)

(State why the correction is necessary.)

When the buffer is not empty, the receiving process can received a message from the buffer directly and this message should be removed from the message queue of this buffer.

6. Submitter (Optional)

(Name, organization, contact information, e.g., phone, email address.)

Yongwang ZHAO, School of Computer Science and Engineering, Beihang University.
Room G506, New main building, Beihang University, Haidian District, Beijing, China.
Phone: +86 10 82339274 Email: zhaoyw@buaa.edu.cn; zhaoyongwang@gmail.com;

Please return comments to fax +1410-266-2047 or standards@arinc.com

Note: Items 2-5 may be repeated for additional errata. All recommendations will be evaluated by the staff. Any substantive changes will require submission to the relevant subcommittee for incorporation into a subsequent Supplement.

[To be completed by IA Staff]

Errata Report Identifier:_____ Engineer Assigned: _____

Review Status: _____