CCSDS Historical Document

This document's Historical status indicates that it is no longer current. It has either been replaced by a newer issue or withdrawn because it was deemed obsolete. Current CCSDS publications are maintained at the following location:

http://public.ccsds.org/publications/

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Practices

SPACECRAFT ONBOARD INTERFACE SERVICES— TIME ACCESS SERVICE

RECOMMENDED PRACTICE

CCSDS 872.0-M-1

MAGENTA BOOK
January 2011

CCSDS HISTORICAL DOCUMENT

The Consultative Committee for Space Data Systems

Recommendation for Space Data System Practices

SPACECRAFT ONBOARD INTERFACE SERVICES— TIME ACCESS SERVICE

RECOMMENDED PRACTICE

CCSDS 872.0-M-1

MAGENTA BOOK
January 2011

AUTHORITY

Issue:	Recommended Practice, Issue 1
Date:	January 2011
Location:	Washington, DC, USA

This document has been approved for publication by the Management Council of the Consultative Committee for Space Data Systems (CCSDS) and represents the consensus technical agreement of the participating CCSDS Member Agencies. The procedure for review and authorization of CCSDS documents is detailed in the *Procedures Manual for the Consultative Committee for Space Data Systems*, and the record of Agency participation in the authorization of this document can be obtained from the CCSDS Secretariat at the address below.

This document is published and maintained by:

CCSDS Secretariat
Space Communications and Navigation Office, 7L70
Space Operations Mission Directorate
NASA Headquarters
Washington, DC 20546-0001, USA

CCSDS HISTORICAL DOCUMENT

STATEMENT OF INTENT

The Consultative Committee for Space Data Systems (CCSDS) is an organization officially established by the management of its members. The Committee meets periodically to address data systems problems that are common to all participants, and to formulate sound technical solutions to these problems. Inasmuch as participation in the CCSDS is completely voluntary, the results of Committee actions are termed **Recommendations** and are not in themselves considered binding on any Agency.

CCSDS Recommendations take two forms: **Recommended Standards** that are prescriptive and are the formal vehicles by which CCSDS Agencies create the standards that specify how elements of their space mission support infrastructure shall operate and interoperate with others; and **Recommended Practices** that are more descriptive in nature and are intended to provide general guidance about how to approach a particular problem associated with space mission support. This **Recommended Practice** is issued by, and represents the consensus of, the CCSDS members. Endorsement of this **Recommended Practice** is entirely voluntary and does not imply a commitment by any Agency or organization to implement its recommendations in a prescriptive sense.

No later than five years from its date of issuance, this **Recommended Practice** will be reviewed by the CCSDS to determine whether it should: (1) remain in effect without change; (2) be changed to reflect the impact of new technologies, new requirements, or new directions; or (3) be retired or canceled.

In those instances when a new version of a **Recommended Practice** is issued, existing CCSDS-related member Practices and implementations are not negated or deemed to be non-CCSDS compatible. It is the responsibility of each member to determine when such Practices or implementations are to be modified. Each member is, however, strongly encouraged to direct planning for its new Practices and implementations towards the later version of the Recommended Practice.

CCSDS HISTORICAL DOCUMENT

FOREWORD

This document is a technical Recommended Practice for use in developing flight and ground systems for space missions and has been prepared by the Consultative Committee for Space Data Systems (CCSDS). The Time Access Service described herein is intended for missions that are cross-supported between Agencies of the CCSDS, in the framework of the Spacecraft Onboard Interface Services (SOIS) CCSDS area.

This Recommended Practice specifies a set of related services to be used by space missions to obtain onboard time. The SOIS Time Access Service provides provides a common service interface and quality of service regardless of the particular type of data link or protocol being used for communication.

Through the process of normal evolution, it is expected that expansion, deletion, or modification of this document may occur. This Recommended Practice is therefore subject to CCSDS document management and change control procedures, which are defined in the *Procedures Manual for the Consultative Committee for Space Data Systems*. Current versions of CCSDS documents are maintained at the CCSDS Web site:

http://www.ccsds.org/

Questions relating to the contents or status of this document should be addressed to the CCSDS Secretariat at the address indicated on page i.

At time of publication, the active Member and Observer Agencies of the CCSDS were:

<u>Member Agencies</u>

- Agenzia Spaziale Italiana (ASI)/Italy.
- Canadian Space Agency (CSA)/Canada.
- Centre National d'Etudes Spatiales (CNES)/France.
- China National Space Administration (CNSA)/People's Republic of China.
- Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)/Germany.
- European Space Agency (ESA)/Europe.
- Instituto Nacional de Pesquisas Espaciais (INPE)/Brazil.
- Japan Aerospace Exploration Agency (JAXA)/Japan.
- National Aeronautics and Space Administration (NASA)/USA.
- Federal Space Agency (FSA)/Russian Federation.
- UK Space Agency/United Kingdom.

<u>Observer Agencies</u>

- Austrian Space Agency (ASA)/Austria.
- Belgian Federal Science Policy Office (BFSPO)/Belgium.
- Central Research Institute of Machine Building (TsNIIMash)/Russian Federation.
- China Satellite Launch and Tracking Control General, Beijing Institute of Tracking and Telecommunications Technology (CLTC/BITTT)/China.
- Chinese Academy of Sciences (CAS)/China.
- Chinese Academy of Space Technology (CAST)/China.
- Commonwealth Scientific and Industrial Research Organization (CSIRO)/Australia.
- CSIR Satellite Applications Centre (CSIR)/Republic of South Africa.
- Danish National Space Center (DNSC)/Denmark.
- Departamento de Ciência e Tecnologia Aeroespacial (DCTA)/Brazil.
- European Organization for the Exploitation of Meteorological Satellites (EUMETSAT)/Europe.
- European Telecommunications Satellite Organization (EUTELSAT)/Europe.
- Geo-Informatics and Space Technology Development Agency (GISTDA)/Thailand.
- Hellenic National Space Committee (HNSC)/Greece.
- Indian Space Research Organization (ISRO)/India.
- Institute of Space Research (IKI)/Russian Federation.
- KFKI Research Institute for Particle & Nuclear Physics (KFKI)/Hungary.
- Korea Aerospace Research Institute (KARI)/Korea.
- Ministry of Communications (MOC)/Israel.
- National Institute of Information and Communications Technology (NICT)/Japan.
- National Oceanic and Atmospheric Administration (NOAA)/USA.
- National Space Agency of the Republic of Kazakhstan (NSARK)/Kazakhstan.
- National Space Organization (NSPO)/Chinese Taipei.
- Naval Center for Space Technology (NCST)/USA.
- Scientific and Technological Research Council of Turkey (TUBITAK)/Turkey.
- Space and Upper Atmosphere Research Commission (SUPARCO)/Pakistan.
- Swedish Space Corporation (SSC)/Sweden.
- United States Geological Survey (USGS)/USA.

DOCUMENT CONTROL

Document	Title	Date	Status
CCSDS 872.0-M-1	Spacecraft Onboard Interface Services—Time Access Service, Recommended Practice, Issue 1	January 2011	Original issue

CONTENTS

CONTENTS (continued)

1 INTRODUCTION

1.1 PURPOSE AND SCOPE OF THIS DOCUMENT

This document is one of a family of documents specifying the Spacecraft Onboard Interface Services (SOIS)-compliant services to be provided in support of onboard applications.

The purpose of this document is to define services and service interfaces provided by the SOIS Time Access Service. Its scope is to specify the service only and not to specify methods of providing the service, although use of the SOIS subnetwork services is assumed.

This document conforms to the principles set out in the SOIS Green Book (reference [A6]) and it is intended to be applied together with it.

1.2 APPLICABILITY

This document applies to any mission or equipment claiming to provide a SOIS-compatible Time Access Service.

1.3 RATIONALE

SOIS provides service interface specifications in order to promote interoperability and development reuse via peer-to-peer and vertical standardisation.

1.4 DOCUMENT STRUCTURE

This document has five major sections:

– this section, containing administrative information, definitions, and references;

– section 2, describing general concepts and assumptions, including security issues;

– section 3, containing the Time Access Service specification;

– section 4, containing the Management Information Base (MIB) for this service;

– section 5, comprising a Service Conformance Statement Proforma.

In addition, four informative annexes are provided:

– annex A, containing a list of informative references;

– annex B, describing how local time registers may be implemented and how the master onboard time value is usually distributed in current spacecraft;

– annex C, describing aspects of the POSIX API time functions which might be useful to implementers of the Time Access Service;

– annex D, describing how an implementation of the service may be characterised.

1.5 CONVENTIONS AND DEFINITIONS

1.5.1 BIT NUMBERING CONVENTION AND NOMENCLATURE

In this document, the following convention is used to identify each bit in an *N*-bit field. The first bit in the field to be transmitted (i.e., the most left justified when drawing a figure) is defined to be ‘Bit 0’; the following bit is defined to be ‘Bit 1’ and so on up to ‘Bit *N*–1’. When the field is used to express a binary value (such as a counter), the Most Significant Bit (MSB) shall be the first transmitted bit of the field, i.e., ‘Bit 0’ (see figure 1-1).

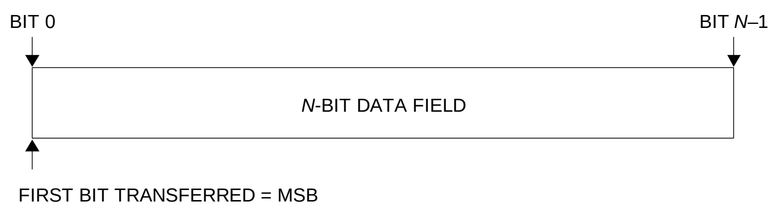

Figure 1-1: Bit Numbering Convention

In accordance with modern data communications practice, spacecraft data fields are often grouped into eight-bit ‘words’ widely known as bytes. Throughout this Recommended Practice, such an eight-bit word is called an ‘octet’.

The numbering for octets within a data structure starts with zero. By CCSDS convention, any ‘spare’ bits shall be permanently set to ‘0’.

1.5.2 DEFINITIONS

1.5.2.1 General

For the purpose of this document the following definitions apply.

1.5.2.2 Definitions from the Open Systems Interconnection (OSI) Basic Reference Model

This document is defined using the style established by the Open Systems Interconnection (OSI) Basic Reference Model (reference [A4]). This model provides a common framework for the development of standards in the field of systems interconnection.

The following terms used in this Recommended Practice are adapted from definitions given in reference [A4]:

layer: subdivision of the architecture, constituted by subsystems of the same rank.

protocol data unit (PDU): unit of data specified in a protocol and consisting of Protocol Control Information (PCI) and possibly user data.

service: capability of a layer (service provider) together with the layers beneath it, which is provided to the service users.

Time Access Service Access Point (TASAP): the point at which Time Access Service is provided by a Time Access Service entity to a Time Access Service user entity.

TASAP address: a Time Access Service address that is used to identify a single TASAP.

1.5.2.3 Terms defined in this Recommended Practice

For the purposes of this Recommended Practice, the following definitions also apply.

accuracy: closeness of the agreement between the times reported by a local time source and the master onboard time source of the spacecraft.

application: component of the onboard software that makes use of the Time Access Service.

NOTE – Such components include flight software applications and higher-layer services.

error bound: indication of the theoretical error range of a particular time value.

NOTES

1 This is dependent upon the mechanisms of the time source used to obtain the time value. It may be related to the precision and resolution of the time value but also any associated synchronisation mechanisms.

2 Larger values for the error bound indicate less accuracy in the returned time value.

precision: smallest significant time increment that can be reported by a time source.

NOTE – For example, a time source that reports time as *minutes:seconds:milliseconds* has a precision of 1ms. The precision of a time source is normally less than the resolution of that time source.

resolution: smallest significant time increment that can be measured by a time source.

NOTE – For example, a time source driven by a 1MHz oscillator has a resolution of 1µs.

time correlation: process of maintaining coherence between the master time source and each local time source, such that all time sources provide the same time value within some bounded uncertainty.

NOTE – This maximizes the accuracy of each time source with respect to the master onboard time source.

1.6 DOCUMENT NOMENCLATURE

The following conventions apply throughout this Recommended Practice:

a) the words ‘shall’ and ‘must’ imply a binding and verifiable specification;

b) the word ‘should’ implies an optional, but desirable, specification;

c) the word ‘may’ implies an optional specification;

d) the words ‘is’, ‘are’, and ‘will’ imply statements of fact.

1.7 REFERENCES

The following documents contain provisions which, through reference in this text, constitute provisions of this Recommended Practice. At the time of publication, the editions indicated were valid. All documents are subject to revision, and users of this Recommended Practice are encouraged to investigate the possibility of applying the most recent editions of the documents indicated below. The CCSDS Secretariat maintains a register of currently valid CCSDS Documents.

NOTE – This document contains no normative references. Informative references are contained in annex A.

2 OVERVIEW

2.1 FUNCTION

The Time Access Service provides a user entity with a consistent interface to a local time source that is correlated to some centrally maintained master onboard time source.

2.2 CONTEXT

The Time Access Service is defined within the context of the overall SOIS architecture (reference [A6]) as one of the services of the Application Support Layer, as illustrated in figure 2-1.

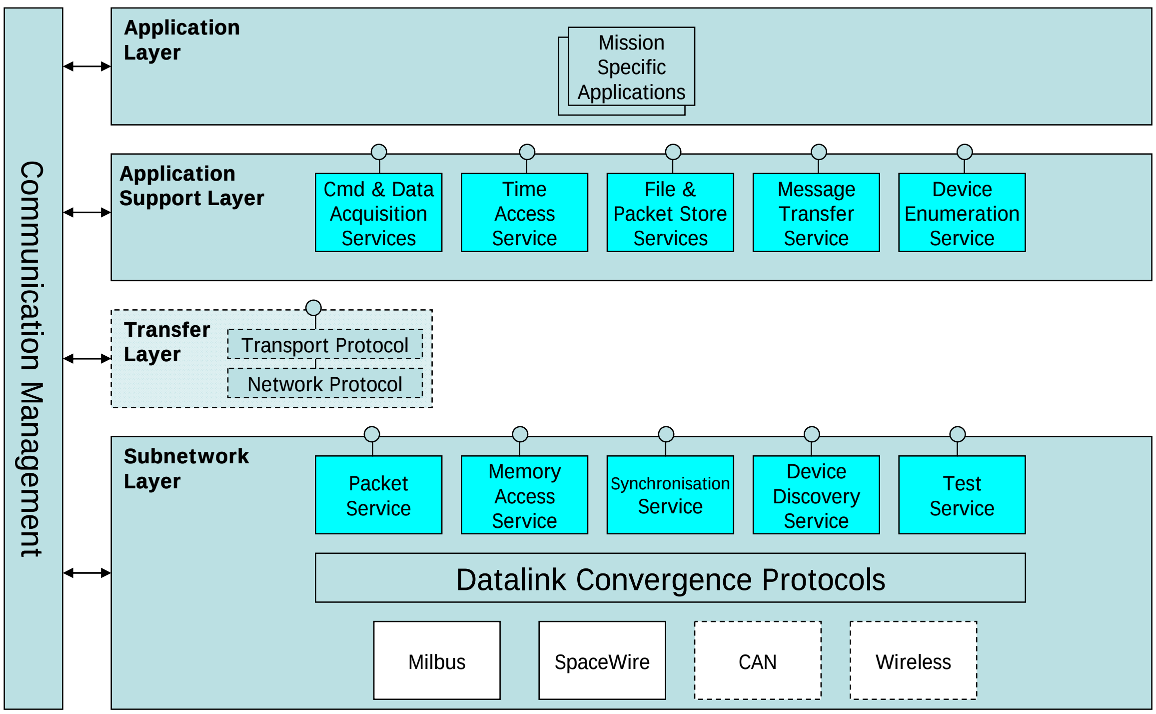

Figure 2-1: Time Access Service Context

The Time Access Service provides applications with a consistent interface to a local time source that is correlated to some centrally maintained master onboard time source. The time values provided by this service might typically be used by the application to schedule some operation, such as the acquisition of an image or to time stamp locally generated telemetry data.

The need to provide a local, correlated time source in onboard processor nodes is common to all spacecraft that have more than one processing node connected to an onboard bus or LAN. A typical architectural scenario is shown in figure 2-2. The Time Access Service is concerned only with providing the interface to the local time source. It is not concerned with the provision of the local time source and mechanisms that may be used to correlate the time between the local and other time sources, the latter of which is addressed by the SOIS Subnetwork Synchronisation Service (reference [A7]).

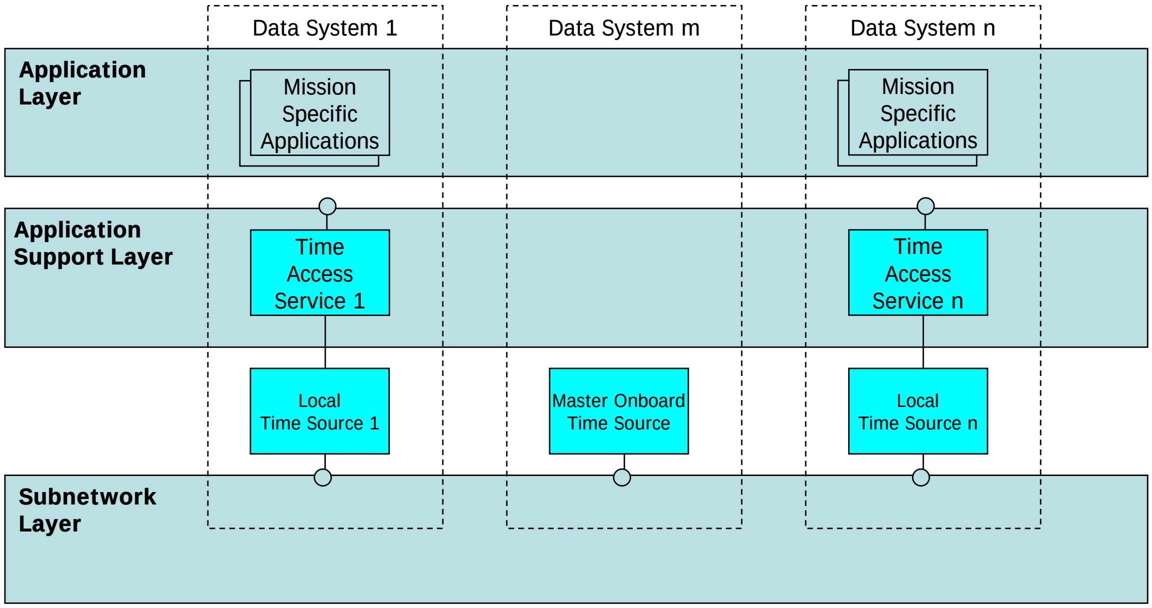

Figure 2-2: Typical Onboard Time System Architecture

NOTE – A typical onboard time system architecture consists of local and master onboard time sources implemented in hardware. The Time Access Service provides access to a local time source only.

In this architecture the local time sources are typically free-running hardware counters accumulating seconds and sub-seconds of elapsed time. Each of these counters is driven by its own oscillator, and the absolute frequency and frequency stability of each oscillator are different in each node. The master onboard time source, the reference for onboard time for all onboard mission operations, is usually a similar free-running counter driven by an oscillator with precise absolute frequency and high stability.[1] The value provided by this time source is usually called the Mission Elapsed Time (MET) or the Spacecraft Elapsed Time (SCET).

The actual method of implementing the master onboard time source and the local time sources and the mechanisms used to correlate them is outside the scope of this document. However, annex B describes a scheme that is typically used today.

This document defines a standard interface between applications hosted on each node and the local time source for that node. The scope of this document is indicated by the dashed box in figure 2-2. The basic capability provided by the Time Access Service is the ability to read the time on demand, i.e., a '*wall clock*' capability. Two optional extensions are also defined

[1] In certain orbits it is possible to use a time source that is correlated to an external reference time source, such as a GPS receiver, as the master onboard time source.

which reflect common requirements for onboard software systems. The first of these is an '*alarm clock*' capability, which enables the application to request notification at a particular time. The second is a '*metronome*' capability, which enables the application to request periodic notifications with a specified interval and starting at a particular time.

The benefit to the user is that all applications have a uniform interface to the local time source, regardless of their location on the spacecraft and how time correlation is actually performed, and do not have to access local hardware directly. This simplifies the development of the applications and means that they can be relocated if necessary and can be re-used in other missions.

2.3 PURPOSE AND OPERATION OF THE TIME ACCESS SERVICE

Applications should use the Time Access Service to obtain the time from the local time source rather than, for example, reading directly from the local elapsed time counter hardware registers. The time provided by the Time Access Service can be used for a variety of purposes, for example, for scheduling of onboard operations and for time stamping of acquired data, possibly using onboard time synchronised across a spacecraft subnetwork.

The Time Access Service is not intended to be used as a multi-purpose timing mechanism for applications requiring specialised hardware; for example, it should not be used for software real-time task scheduling.

From the application software perspective, use of the Time Access Service will result in applications that are more portable, easier to develop, and independent of the hardware implementations of the onboard time sources. From the spacecraft platform implementers' perspective, use of the Time Access Service will make it easier to control the access to shared hardware resources.

The Time Access Service is operated using service requests and service indications passed between the service user and the service provider.

2.4 SECURITY

2.4.1 SECURITY BACKGROUND

The SOIS services are intended for use with protocols that operate solely within the confines of an onboard subnet. It is therefore assumed that SOIS services operate in an isolated environment which is protected from external threats. Any external communication is assumed to be protected by services associated with the relevant space-link protocols. The specification of such security services is outside the scope of this document.

2.4.2 SECURITY CONCERNS

At the time of writing there are no identified security concerns. If confidentiality of data is required within a spacecraft it is assumed it is applied at the Application layer. Reference [A8] contains more information regarding the choice of service and where it can be implemented.

2.4.3 POTENTIAL THREATS AND ATTACK SCENARIOS

Potential threats and attack scenarios typically derive from external communication and are therefore not the direct concern of the SOIS services, which make the assumption that the services operate within a safe and secure environment. It is assumed that all applications executing within the spacecraft have been thoroughly tested and cleared for use by the mission implementer. Confidentiality of applications can be provided by Application layer mechanisms or by specific implementation methods such as time and space partitioning. Such methods are outside the scope of SOIS.

2.4.4 CONSEQUENCES OF NOT APPLYING SECURITY

The security services are outside the scope of this document and are expected to be applied at layers above or below those specified in this document. If confidentiality is not implemented, science data or other parameters transmitted within the spacecraft might be visible to other applications resident within the spacecraft, resulting in disclosure of sensitive or private information.

3 TIME ACCESS SERVICE

3.1 PROVIDED SERVICE

3.1.1 GENERAL

a) The Time Access Service shall implement the wall clock capability specified in 3.1.2.

b) The Time Access Service may implement the alarm clock capability specified in 3.1.3.

c) The Time Access Service may implement the metronome capability specified in 3.1.4.

3.1.2 WALL CLOCK CAPABILITY

a) The wall clock capability shall allow the user to obtain the local onboard time on demand.

b) To obtain the time, the user shall issue a TIME.request primitive.

c) The service shall respond to a TIME.request with a TIME.indication primitive.

NOTE – As specified in 3.4.3, this primitive contains the outcome of obtaining the local onboard time, the obtained current local time, and an indication of the accuracy and validity of this time value.

3.1.3 ALARM CLOCK CAPABILITY (OPTIONAL)

a) If implemented, the alarm clock capability shall allow the user to register for an alarm call at a specified time.

b) To register for an alarm call, the user shall issue an ALARM.request primitive with the time (absolute or relative) at which the alarm call is to be issued.

c) The service shall respond to an ALARM.request immediately with an ALARM.indication indicating the outcome of the registration and, for a successful registration, a TIME.indication primitive issued at the specified time.

d) The user may cancel a pending alarm call by issuing a CANCEL_ALARM.request.

e) The service shall respond to a CANCEL_ALARM.request immediately with a CANCEL_ALARM.indication indicating the outcome of the cancellation.

3.1.4 METRONOME CAPABILITY (OPTIONAL)

a) If implemented, the metronome capability shall allow the user to register for a periodic indication of the time at a specified frequency and starting at a specific time.

b) To register for a periodic indication of the time, the user shall issue a METRONOME.request primitive with the time at which the first periodic indication is to be issued and the frequency of subsequent indications.

c) The service shall respond to a METRONOME.request immediately with a METRONOME.indication indicating the outcome of the registration and, for a successful registration, periodic TIME.indication primitives with the first indication being issued at the specified start time and subsequent indications being issued at the specified frequency.

d) To stop the metronome, the user shall issue a CANCEL_METRONOME.request.

e) The service shall respond to a CANCEL_METRONOME.request immediately with a CANCEL_METRONOME.indication indicating the outcome of the cancellation.

3.2 DISCUSSION: EXPECTED SERVICES FROM UNDERLYING LAYERS

The expected service from the underlying layers is a locally maintained clock that indicates seconds and sub-seconds of monotonically increasing elapsed time.

This does not preclude the locally maintained clock's being 'wound back', e.g., because of an inaccuracy causing it to run fast. This may be expressed as an elapsed time from some specific epoch, such as the start of the mission, or may be a real-time clock. The maximum adjustment permissible per request will be defined in the Time Access Service's MIB, so as to minimise the affect upon the service users.

3.3 SERVICE PARAMETERS

3.3.1 GENERAL

The Time Access Service shall use the parameters specified in 3.3.2 to 3.3.11.

3.3.2 TIME ACCESS SERVICE ACCESS POINT ADDRESS

The Time Access Service Access Point (TASAP) parameter shall identify the SAP Address that identifies the user entity that wishes to invoke the Time Access Service.

3.3.3 TRANSACTION IDENTIFIER

a) The Transaction Identifier parameter shall be a value, assigned by the invoking user entity, which is subsequently used to associate indication primitives with the causal request primitives.

NOTE – The user entity is thus able to correlate all indications and confirmations with the originating service request.

b) Transaction Identifier shall be unique within the user entity.

c) Uniqueness in the service provider shall be achieved through the concatenation of TASAP Address and Transaction Identifier.

3.3.4 ALARM IDENTIFIER

a) The Alarm Identifier parameter shall be a value, assigned by the service provider, which is subsequently used to associate request primitives with a created alarm.

b) Alarm Identifier shall be unique within a service provider.

3.3.5 METRONOME IDENTIFIER

a) The Metronome Identifier parameter shall be a value, assigned by the service provider, which is subsequently used to associate request primitives with a created alarm.

b) Metronome Identifier shall be unique within a service provider.

3.3.6 RESULT METADATA

The Result Metadata parameter shall be used to provide information generated by the Time Access Service provider to the service invoking entity to provide information related to the successful or failed result of a time access operation.

NOTE – The parameter can also include other information indicating failure conditions, e.g., the specified request could not be serviced within the managed timeout period or the Time Access Service is not functioning correctly.

3.3.7 CURRENT TIME

The Current Time parameter shall provide the current onboard time as determined by the Time Access Service provider.

3.3.8 CURRENT TIME ERROR SPECIFICATION

The Current Time Error Specification parameter shall indicate the error of the associated Current Time parameter as determined by the Time Access Service provider.

3.3.9 ALARM-AT TIME

The Alarm-At Time parameter shall indicate the time at which the user wishes to receive a time indication.

3.3.10 FIRST ALARM TIME

The First Alarm Time parameter shall indicate the time at which the user wishes to receive the first of a series of periodic alarms.

3.3.11 INTER-ALARM INTERVAL

The Inter-Alarm Interval parameter shall indicate the requested inter-alarm time interval between successive alarms.

3.4 TIME ACCESS SERVICE PRIMITIVES

3.4.1 GENERAL

a) The Time Access Service interface shall implement the following primitives:

   1) TIME.request, as specified in 3.4.2.

   2) TIME.indication, as specified in 3.4.3.

b) The Time Access service interface may implement the following primitives:

   1) ALARM.request, as specified in 3.4.4.

   2) ALARM.indication, as specified in 3.4.5.

   3) CANCEL_ALARM.request, as specified in 3.4.6.

   4) CANCEL_ALARM.indication, as specified in 3.4.7.

   5) METRONOME.request, as specified in 3.4.8.

   6) METRONOME.indication, as specified in 3.4.9.

   7) CANCEL_METRONOME.request, as specified in 3.4.10.

   8) CANCEL_METRONOME.indication, as specified in 3.4.11.

3.4.2 TIME.REQUEST

3.4.2.1 Function

The **TIME.request** shall be used to request the service to retrieve the current time.

3.4.2.2 Semantics

The **TIME.request** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

TIME.request (TASAP Address, Transaction Identifier)

3.4.2.3 When Generated

The **TIME.request** shall be passed to the Time Access Service provider to request the current time.

3.4.2.4 Effect on Receipt

Receipt of the **TIME.request** primitive shall cause the Time Access Service provider to determine the current time.

3.4.2.5 Additional Comments

None.

3.4.3 TIME.INDICATION

3.4.3.1 Function

The **TIME.indication** primitive shall be used to pass the current time to the user entity.

3.4.3.2 Semantics

The **TIME.indication** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

> **TIME.indication** (TASAP Address, Transaction Identifier, Current Time, Current Time Error Specification, Result Metadata)

3.4.3.3 When Generated

The **TIME.indication** primitive shall be issued by the service provider to the receiving user entity in response to a TIME.request, ALARM.request, or a METRONOME.request.

NOTE – This primitive provides the current time (see 3.3.7 and 3.3.8), and metadata concerning whether the request was executed successfully or not (see 3.3.4).

3.4.3.4 Effect on Receipt

The response of the user entity to a **TIME.indication** primitive is unspecified.

3.4.3.5 Additional Comments

None.

3.4.4 ALARM.REQUEST

3.4.4.1 Function

The **ALARM.request** primitive shall be used to request the service create a one-shot alarm in the form of a time indication at a specific time in the future.

3.4.4.2 Semantics

The **ALARM.request** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

ALARM.request (TASAP Address, Transaction Identifier, Alarm-At Time)

3.4.4.3 When Generated

The **ALARM.request** primitive shall be passed to the Time Access Service provider in order to request a one-shot alarm be created.

3.4.4.4 Effect on Receipt

Receipt of the **ALARM.request** primitive shall cause the Time Access Service provider to create a one-shot alarm.

3.4.4.5 Additional Comments

None.

CCSDS HISTORICAL DOCUMENT

3.4.5 ALARM.INDICATION

3.4.5.1 Function

The **ALARM.indication** primitive shall be used to indicate the outcome of creating a one-shot alarm to the user entity.

3.4.5.2 Semantics

The **ALARM.indication** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

> **ALARM.indication** (TASAP Address, Transaction Identifier, Alarm Identifier, Result Metadata)

3.4.5.3 When Generated

The **ALARM.indication** primitive shall be issued by the service provider to the receiving user entity in response to an ALARM.request.

NOTE – This primitive provides metadata concerning whether the request was executed successfully or not (see 3.4.4).

3.4.5.4 Effect on Receipt

The response of the user entity to an **ALARM.indication** primitive is unspecified.

3.4.5.5 Additional Comments

None.

3.4.6 CANCEL_ALARM.REQUEST

3.4.6.1 Function

The **CANCEL_ALARM.request** primitive shall be used to request the service cancel a previously requested alarm.

3.4.6.2 Semantics

The **CANCEL_ALARM.request** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

> **CANCEL_ALARM.request** (TASAP Address, Transaction Identifier, Alarm Identifier)

3.4.6.3 When Generated

The **CANCEL_ALARM.request** primitive shall be passed to the Time Access Service provider to request that a previously requested alarm be cancelled.

3.4.6.4 Effect on Receipt

Receipt of the **CANCEL_ALARM.request** primitive shall cause the Time Access Service provider to cancel the scheduled alarm and remove all state associated with it.

3.4.6.5 Additional Comments

None.

3.4.7 CANCEL_ALARM.INDICATION

3.4.7.1 Function

The **CANCEL_ALARM.indication** primitive shall be used to indicate the outcome of cancelling a one-shot alarm to the user entity.

3.4.7.2 Semantics

The **CANCEL_ALARM.indication** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

> **CANCEL_ALARM.indication** (TASAP Address, Transaction Identifier, Result Metadata)

3.4.7.3 When Generated

The **CANCEL_ALARM.indication** primitive shall be issued by the service provider to the receiving user entity in response to a CANCEL_ALARM.request.

NOTE – This primitive provides metadata concerning whether the request was executed successfully or not (see 3.4.6).

3.4.7.4 Effect on Receipt

The response of the user entity to a **CANCEL_ALARM.indication** primitive is unspecified.

3.4.7.5 Additional Comments

None.

3.4.8 METRONOME.REQUEST

3.4.8.1 Function

The **METRONOME.request** primitive shall be used to request the service create a metronome in the form of a periodic time indication at a specific time in the future and at specified intervals thereafter.

3.4.8.2 Semantics

The **METRONOME.request** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

> **METRONOME.request** (TASAP Address, Transaction Identifier, First Alarm Time, Inter-Alarm Interval)

3.4.8.3 When Generated

The **METRONOME.request** primitive shall be passed to the Time Access Service provider to request the creation of a metronome.

3.4.8.4 Effect on Receipt

Receipt of the **METRONOME.request** primitive shall cause the Time Access Service provider to create the metronome.

3.4.8.5 Additional Comments

None.

3.4.9 METRONOME.INDICATION

3.4.9.1 Function

The **METRONOME.indication** primitive shall be used to indicate the outcome of creating a metronome to the user entity.

3.4.9.2 Semantics

The **METRONOME.indication** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

> **METRONOME.indication** (TASAP Address, Transaction Identifier, Metronome Identifier, Result Metadata)

3.4.9.3 When Generated

The **METRONOME.indication** primitive shall be issued by the service provider to the receiving user entity in response to a METRONOME.request.

NOTE – This primitive provides metadata concerning whether the request was executed successfully or not (see 3.4.8).

3.4.9.4 Effect on Receipt

The response of the user entity to a **METRONOME.indication** primitive is unspecified.

3.4.9.5 Additional Comments

None.

3.4.10 CANCEL_METRONOME.REQUEST

3.4.10.1 Function

The **CANCEL_METRONOME.request** primitive shall be used to request the service cancel a previously requested metronome.

3.4.10.2 Semantics

The **CANCEL_METRONOME.request** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

> **CANCEL_METRONOME.request** (TASAP Address, Transaction Identifier, Metronome Identifier)

3.4.10.3 When Generated

The **CANCEL_METRONOME.request** primitive shall be passed to the Time Access Service provider to request that a previously requested metronome be cancelled.

3.4.10.4 Effect on Receipt

The **CANCEL_METRONOME.request** primitive shall cause the Time Access Service provider to cancel the metronome and remove all state associated with it.

3.4.10.5 Additional Comments

None.

3.4.11 CANCEL_METRONOME.INDICATION

3.4.11.1 Function

The **CANCEL_METRONOME.indication** primitive shall be used to indicate the outcome of cancelling a metronome to the user entity.

3.4.11.2 Semantics

The **CANCEL_METRONOME.indication** primitive shall use the following semantics, with the meaning of the parameters specified in 3.3.

> **CANCEL_METRONOME.indication** (TASAP Address, Transaction Identifier, Result Metadata)

3.4.11.3 When Generated

The **CANCEL_METRONOME.indication** shall be issued by the service provider to the receiving user entity in response to a CANCEL_METRONOME.request.

NOTE – This primitive provides metadata concerning whether the request was executed successfully or not (see 3.4.10).

3.4.11.4 Effect on Receipt

The response of the user entity to a **CANCEL_METRONOME.indication** primitive is unspecified.

3.4.11.5 Additional Comments

None.

CCSDS HISTORICAL DOCUMENT

4 MANAGEMENT INFORMATION BASE

4.1 GENERAL

There is currently no Management Information Base associated with this service. All management items are associated with the protocol providing the service. However, guidance is provided as to MIB contents in section 4.3.

4.2 SPECIFICATIONS

Any protocol claiming to provide this service in a SOIS-compliant manner shall publish its Management Information Base as part of the protocol specification.

4.3 MIB GUIDANCE

The MIB of the protocol providing the Time Access Service should consider the following aspects:

a) Frequency, as specified in 4.4.

b) Drift, as specified in 4.5.

c) Maximum adjustment, as specified in 4.6.

NOTE – These aspects are not in any way an indication of the complete contents of a MIB for a protocol providing the Time Access Service but are offered as guidance as to those aspects of the MIB which may relate to the Time Access Service interface.

4.4 FREQUENCY

4.4.1 The **Frequency** parameter shall indicate the rate, in Hertz, at which the Time Access Service is updated.

NOTES

1 This parameter is not the frequency of a local time source, e.g., hardware clock. It is the frequency at which the Time Access Service samples it.

2 This parameter is defined in terms of frequency rather than interval. While this is inconsistent with the rest of the document, the terminology is retained since it is more usual to refer to the frequency of a local time source.

4.4.2 A service management entity should be able to access the frequency parameter.

NOTE – However, it is usually not possible to update it when the local time source is implemented in hardware.

4.5 DRIFT

4.5.1 The **Drift** parameter shall indicate the stability of the local oscillator used in the local clock.

4.5.2 A service management entity should be able to access the drift parameter.

4.5.3 A service management entity should be able to update the drift parameter to compensate for clock aging effects.

4.6 MAXIMUM ADJUSTMENT

The **Maximum Adjustment** parameter shall indicate the maximum permissible adjustment, forward or backward, that can be made to the local time source resident within the spacecraft.

5 SERVICE CONFORMANCE STATEMENT PROFORMA

For any implementation claiming to provide the Time Access Service, the proforma in table 5-1 shall be completed, giving details of the capabilities of the specification, and made available to any party evaluating the use of the implementation to which the completed proforma refers.

Table 5-1: Service Conformance Statement Proforma

Service Conformance Statement Time Access Service

Implementation Information

Protocol Specification Identification	
Version	
Underlying Services	

Mandatory Features

TIME.request	√
TIME.indication	√

Optional Features

ALARM.request	
ALARM.indication	
CANCEL_ALARM.request	
CANCEL_ALARM.indication	
METRONOME.request	
METRONOME.indication	
CANCEL_METRONOME.request	
CANCEL_METRONOME.indication	

Other Information

N/A	

ANNEX A

INFORMATIVE REFERENCES

(INFORMATIVE)

[A1] *Procedures Manual for the Consultative Committee for Space Data Systems*. CCSDS A00.0-Y-9. Yellow Book. Issue 9. Washington, D.C.: CCSDS, November 2003.

[A2] *Information Technology—Portable Operating System Interface (POSIX)—Part 1: Base Definitions*. International Standard, ISO/IEC 9945-1:2003. 4th ed. Geneva: ISO, 2003.

[A3] *Time Code Formats*. Recommendation for Space Data System Standards, CCSDS 301.0-B-3. Blue Book. Issue 3. Washington, D.C.: CCSDS, January 2002.

[A4] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*. International Standard, ISO/IEC 7498-1. 2nd ed. Geneva: ISO, 1994.

[A5] *The Open Group Base Specifications*. Issue 6. IEEE Std 1003.1, 2004 Edition. San Francisco and Piscataway, NJ: The Open Group and IEEE, 2004.

[A6] *Spacecraft Onboard Interface Services*. Report Concerning Space Data System Standards, CCSDS 850.0-G-1. Green Book. Issue 1. Washington, D.C.: CCSDS, June 2007.

[A7] *Spacecraft Onboard Interface Services—Subnetwork Synchronisation Service*. Recommendation for Space Data System Standards, CCSDS 853.0-M-1. Magenta Book. Issue 1. Washington, D.C.: CCSDS, December 2009.

[A8] *The Application of CCSDS Protocols to Secure Systems*. Report Concerning Space Data System Standards, CCSDS 350.0-G-2. Green Book. Issue 2. Washington, D.C.: CCSDS, January 2006.

NOTE – Normative references are listed in 1.7.

ANNEX B

TYPICAL IMPLEMENTATION

(INFORMATIVE)

This annex describes a hardware-based implementation scheme for local time sources that is commonly used in modern spacecraft. This description is provided for information only.

A typical local time source implementation comprises a free-running counter clocked by a local crystal oscillator. Typically, the oscillator frequency and counter configuration are chosen such that the elapsed time accumulated by the counter is related to an exact number of seconds and sub-seconds. For example, a 32-bit counter might be used with 24-bits corresponding to whole seconds and eight-bits of sub-seconds. This time source would have a precision of just under 4ms and a roll-over period of about 194 days. This time source format is convenient for scheduling onboard operations and also matches the CCSDS Unsegmented Time Code (CUC) format often used for telemetry packet time stamping (see reference [A3]).

Left to its own devices this local time source would record the elapsed time since the unit was powered on. However, the local oscillator has particular frequency accuracy and drift characteristics that will differ from other local time source oscillators on the spacecraft. Furthermore, the flight units may be powered on at different times during the mission. Consequently, each node will report a different value of elapsed time according to the local parameters. Therefore there is usually a requirement to select a single onboard time source as the reference source, or master, for the mission elapsed time, and to implement a mechanism that correlates all of the local time sources so that they report the same time value within reasonable bounds determined by the mission requirements.

Typically this is done by periodically broadcasting the reference source time value via the spacecraft data handling bus and issuing a precisely timed reference pulse indicating exactly when this time is valid. A common implementation, for example, is to broadcast the whole seconds value of the reference mission elapsed time value on a MIL-STD-1553B bus and distribute a dedicated pulse-per-second (PPS) signal to all nodes indicating when this time becomes valid. This is analogous to the talking clock service: ‘at the first stroke it will be 10:58 precisely’.

The way this is handled in the local time source implementation is to add a special *datation* register that is loaded with the broadcast time value received on the spacecraft bus as shown in figure B-1**.** This value is then transferred into the whole seconds’ section of the local elapsed time counter on receipt of the PPS signal. At the same instant the sub-seconds portion of the local counter is cleared. Thus the local time source is accurately correlated to the reference mission elapsed time exactly once per second. Following this instance, the local time source will drift according to the characteristics of the local oscillator until, just before the next PPS signal, it can be expected to reach its maximum deviation with respect to this reference.

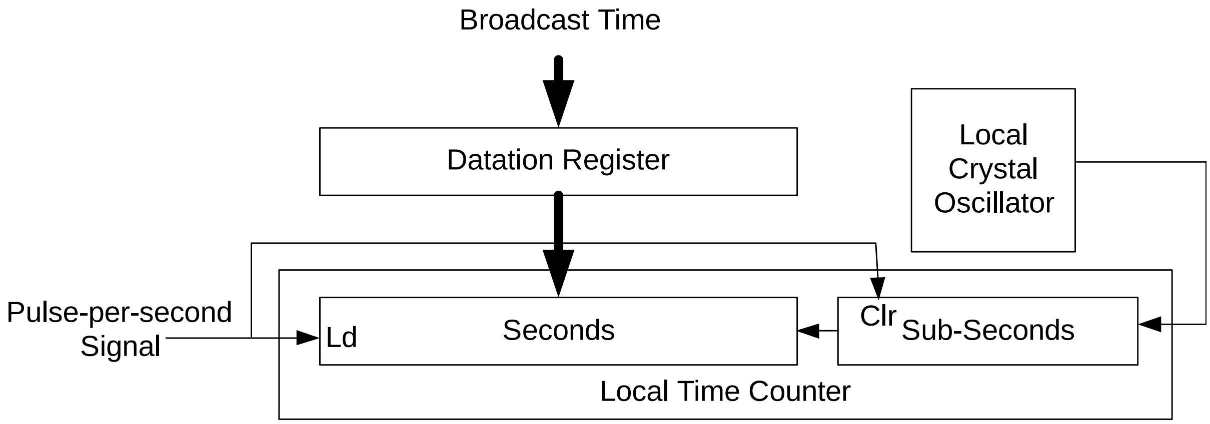

Figure B-1: Typical Local Time Source Implementation

If the local time source is sufficiently sophisticated, it can detect missed correlation steps and can be configured such that the correlation operation never causes the indicated time to run backwards. These techniques are beyond the scope of this brief description.

ANNEX C

TYPICAL API – POSIX

(INFORMATIVE)

NAME[2]

time.h - time types

SYNOPSIS

#include <time.h>

DESCRIPTION

Some of the functionality described on this reference page extends the ISO C standard. Applications shall define the appropriate feature test macro (see the System Interfaces volume of IEEE Std 1003.1-2001, Section 2.2, The Compilation Environment) to enable the visibility of these symbols in this header.

The <time.h> header shall declare the structure tm, which shall include at least the following members:

int tm_sec Seconds [0,60].

int tm_min Minutes [0,59].

int tm_hour Hour [0,23].

int tm_mday Day of month [1,31].

int tm_mon Month of year [0,11].

int tm_year Years since 1900.

int tm_wday Day of week [0,6] (Sunday =0).

int tm_yday Day of year [0,365].

int tm_isdst Daylight Savings flag.

The value of tm_isdst shall be positive if Daylight Savings Time is in effect, 0 if Daylight Savings Time is not in effect, and negative if the information is not available.

The <time.h> header shall define the following symbolic names:

NULL

Null pointer constant.

CLOCKS_PER_SEC

A number used to convert the value returned by the clock() function into seconds.

CLOCK_PROCESS_CPUTIME_ID

The identifier of the CPU-time clock associated with the process making a clock() or timer*() function call.

CLOCK_THREAD_CPUTIME_ID

The identifier of the CPU-time clock associated with the thread making a clock() or timer*() function call.

The <time.h> header shall declare the structure timespec, which has at least the following members:

time_t tv_sec Seconds.

long tv_nsec Nanoseconds.

The <time.h> header shall also declare the itimerspec structure, which has at least the following members:

[2] [6] Copyright © 2001-2004 The IEEE and The Open Group, All Rights reserved

struct timespec it_interval Timer period.

struct timespec it_value Timer expiration.

The following manifest constants shall be defined:

CLOCK_REALTIME

The identifier of the system-wide realtime clock.

TIMER_ABSTIME

Flag indicating time is absolute. For functions taking timer objects, this refers to the clock associated with the timer.

CLOCK_MONOTONIC

The identifier for the system-wide monotonic clock, which is defined as a clock whose value cannot be set via clock_settime() and which cannot have backward clock jumps. The maximum possible clock jump shall be implementation-defined.

The clock_t, size_t, time_t, clockid_t, and timer_t types shall be defined as described in <sys/types.h>.

Although the value of CLOCKS_PER_SEC is required to be 1 million on all XSI-conformant systems, it may be variable on other systems, and it should not be assumed that CLOCKS_PER_SEC is a compile-time constant.

The <time.h> header shall provide a declaration for getdate_err.

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided.

```
char  *asctime(const struct tm *);
char  *asctime_r(const struct tm *restrict, char *restrict);
clock_t clock(void);
int  clock_getcpuclockid(pid_t, clockid_t *);
int  clock_getres(clockid_t, struct timespec *);
int  clock_gettime(clockid_t, struct timespec *);
int  clock_nanosleep(clockid_t, int, const struct timespec *,
    struct timespec *);
int  clock_settime(clockid_t, const struct timespec *);
char  *ctime(const time_t *);
char  *ctime_r(const time_t *, char *);
double  difftime(time_t, time_t);
struct tm *getdate(const char *);
struct tm *gmtime(const time_t *);
struct tm *gmtime_r(const time_t *restrict, struct tm *restrict);
struct tm *localtime(const time_t *);
struct tm *localtime_r(const time_t *restrict, struct tm *restrict);
time_t  mktime(struct tm *);
int  nanosleep(const struct timespec *, struct timespec *);
size_t  strftime(char *restrict, size_t, const char *restrict,
   const struct tm *restrict);
char  *strptime(const char *restrict, const char *restrict,
    struct tm *restrict);
```

```
time_t  time(time_t *);
int  timer_create(clockid_t, struct sigevent *restrict,
   timer_t *restrict);
int  timer_delete(timer_t);
int  timer_gettime(timer_t, struct itimerspec *);
int  timer_getoverrun(timer_t);
int  timer_settime(timer_t, int, const struct itimerspec *restrict,
   struct itimerspec *restrict);
void  tzset(void);
```

The following shall be declared as variables:

```
extern int daylight;
extern long timezone;
extern char *tzname[];
```

Inclusion of the <time.h> header may make visible all symbols from the <signal.h> header.

Mapping of Time Access Service onto POSIX

Time Access Service Primitive	**POSIX API**	**Comments**
Wall Clock Capability		
TIME.request and TIME.indication	clock_gettime()	Recommended. It is also recommended to use the realtime clock, i.e. CLOCK_REALTIME
	time()	Not recommended
Alarm Clock Capability - Blocking		
ALARM.request and TIME.indication	clock_nanosleep()	Recommended. It is also recommended to use the realtime clock, i.e. CLOCK_REALTIME
	nanosleep()	Recommended
	sleep()	Not recommended
Alarm Clock Capability - Callback		
ALARM.request	timer_create(), timer_settime()	Create timer to signal when time expires
CANCEL_ALARM.request	timer_delete()	
TIME.indication	Callback function defined using sigaction()	sigaction() sets the action performed upon a signal occurring, i.e., define callback function
Metronome Capability		
METRONOME.request	Not supported	
CANCEL_METRONOME.request	Not supported	

ANNEX D

SERVICE CHARACTERIZATION

(INFORMATIVE)

This annex presents a minimal set of tests needed to characterize the performances of the service within the context of an integrated HW/SW system. This description is provided for information only.

Service performance for Time Access Service is particularly important because of the intrinsic time dependency between the value of time returned by the service and the time when the value is actually delivered to the requesting user application.

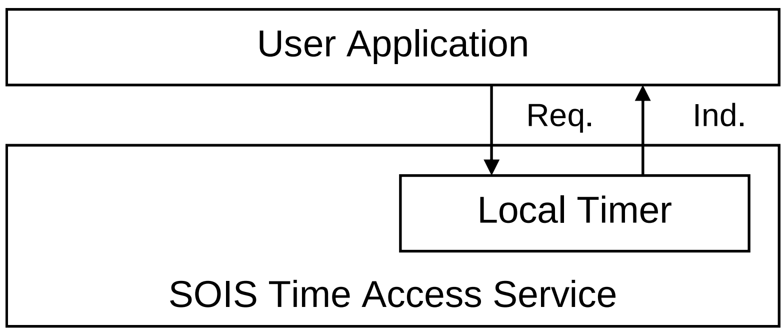

Figure D-1: Time Access Service Abstraction

As presented in figure D-1, Time Access Service provide a means to the service user to retrieve time (and time related services) as if the timer were directly accessible and ideally with zero delay between the time event and the indication to the requesting user.

In practice, as also exemplified in annexes B and C, the local timer is accessed using RTOS or Basic Software services and the call sequence will introduce a delay between the actual time event and the indication to the user.

Figure D-2 presents an example of a possible deployment of Time Access Service main functions on a real system.

This time delay requires the integrated system to be characterized, since the overall performance depends on HW features, on RTOS, and Time Access Service services' implementation.

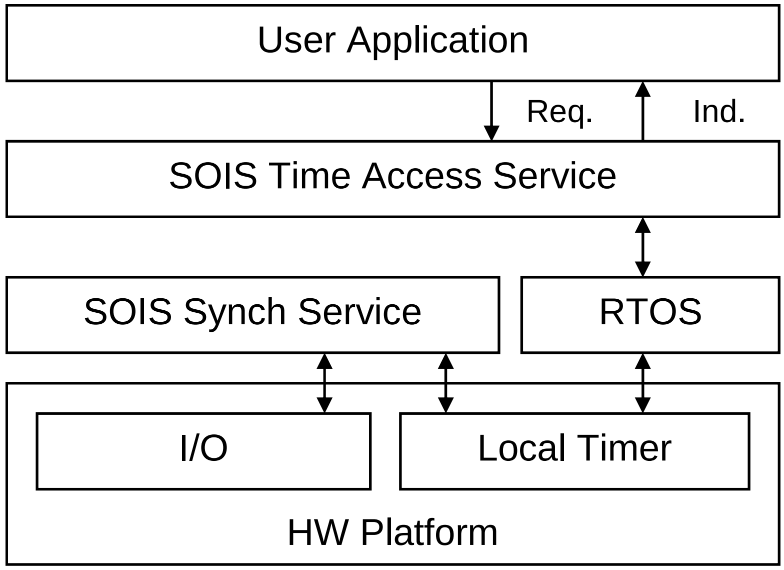

Figure D-2: Typical Deployment of Time Access Service on a Real System

In the frame of the validation of a SOIS-based system, testing and time characterization of the following Time Access Service attributes should be foreseen (as minimum):

- maximum time between TIME.request and corresponding TIME.indication;
- maximum time between alarm or metronome event occurrences and user notification via TIME.indication.

It is important to verify the values obtained from the characterization in time against each real system's system performance requirements.