timex:1.1

API Reference

X Platform, X5 HEVC SDI 1.0.2

23/04/2026

appear

TABLE OF CONTENTS

1 Overview

Overview

Timex deals with topics like NTP, PTP, manual time, system time, FPGA time, time distribution to line cards, in-band PTP etc.

Timex 1.1 offers a set of two JSON RPC APIs:

- `systemTimeSettings` - deals with system wide time configuration
- `cardPtp` - deals with PTP via card's own data ports, aka in-band PTP

See respective documentation sections.

History

`systemTimeSettings` JSON RPC API had previously been part of `mmi` module

- `timex.1.0/systemTimeSettings/...` is equivalent of `mmi.4.7/systemTimeSettings/...`
- Timex 1.1 and newer is not available under `mmi` name

Changelog

1.1

Added

- added `cardPtp` 1.0 API for cards which are capable of getting PTP time from its own data ports, aka in-band PTP

Changed

- `systemTimeSettings` from 1.4 to 1.5

2 cardPtp (1.0)

2.1 Overview

Overview

This module contains JSON RPC API which:

- deals with in-band PTP (PTP via card's own data ports)
- available on line cards which support in-band PTP
- URL: `https://[IP ADDRESS]/board/[SLOT]/api/jsonrpc`
- method names: `timex:1.1/cardPtp/[METHOD NAME]`

2.2 Command Reference

2.2.1 SetPtpSettings

- **message SetPtpSettings.Request**
- **message SetPtpSettings.Response**

2.2.2 GetPtpSettings

- **message GetPtpSettings.Request**
- **message GetPtpSettings.Response**

2.2.3 GetPtpStatus

- **message GetPtpStatus.Request**
- **message GetPtpStatus.Response**

2.2.4 GetIpInterfaces

Returns a collection of IP interfaces that can be used for PTP.

- **message GetIpInterfaces.Request**
- **message GetIpInterfaces.Response**

2.2.5 GetCardPtpCapabilities

- **message GetCardPtpCapabilities.Request**
- **message GetCardPtpCapabilities.Response**

2.3 Type Reference

2.3.1 CardPtpSettings

-

struct

`inBandPtpEnabled`	**bool** Set to true to get PTP time synchronization via this card's IP interfaces. To use in-band PTP a card must also be configured to use PTP as time source using product specific API. When a card is configured to use PTP (via product specific API) and in-band PTP is enabled with this API, only then in-band PTP will work. •
`config`	**ptpTypes.PtpSettings** Used when in-band PTP is enabled. Otherwise it is only stored for future use when in-band PTP gets enabled.

2.3.2 CardPtpStatus

struct

`ptp`	**ptpTypes.PtpStatus**
`activePort`	**optional ipTypes.IpInterfaceId**
`configDiagnostics`	**ptpTypes.PtpConfigDiagnostics**

2.3.3 GetCardPtpCapabilities.Request

empty **struct**

2.3.4 GetCardPtpCapabilities.Response

struct

`caps`	**ptpTypes.CardPtpCapabilities**

2.3.5 GetIpInterfaces.Request

empty **struct**

2.3.6 GetIpInterfaces.Response

struct

`interfaces`	**list** of **ipTypes.IpInterface**

2.3.7 GetPtpSettings.Request

empty **struct**

2.3.8 GetPtpSettings.Response

CardPtpSettings

2.3.9 GetPtpStatus.Request

empty **struct**

2.3.10 GetPtpStatus.Response

struct

`status`	**optional** **CardPtpStatus**

2.3.11 SetPtpSettings.Request

CardPtpSettings

2.3.12 SetPtpSettings.Response

empty **struct**

3 ipTypes (1.0)

3.1 Type Reference

3.1.1 IpInterface

struct

id	**IpInterfaceId**
isEnabled	**bool**
ipVersion	**IpProtocolVersion**

3.1.2 IpInterfaceId

struct

physicalPort	**PhysicalPort**
subInterface	**optional int**
vlanTag	**optional int**

3.1.3 IpProtocolVersion

Describes which version(s) of IP protocol are enabled for an IP interface.

enum

None
V4
V6
Both

3.1.4 PhysicalPort

enum

Unknown
CTRL
D1
D2
D3
D4

4 ntpTypes (1.0)

4.1 Type Reference

4.1.1 NtpServer

struct

ipAddress	**string**

4.1.2 NtpSettings

struct

servers	**list** of **NtpServer**

4.1.3 NtpStatus

struct

inSync	**bool**
server	**string**
stratum	**int**
accuracy	**float**
frequency	**float**
offset	**float**
clkJitter	**float**
sysJitter	**float**

5 ptpTypes (1.0)

5.1 Type Reference

5.1.1 CardPtpCapabilities

struct

inBandPtpAllowed	**bool**
allowedTransportProtocols	**set** of **TransportProtocol**
allowedTransportModes	**set** of **TransportMode**
physicalPorts	**set** of **ipTypes.PhysicalPort**
vlanSupported	**bool**
hybridModeSupported	**bool**

5.1.2 DelayMechanism

enum

E2E
P2P
Auto

5.1.3 Endpoint

struct

address	**string**

5.1.4 FloatProfileSetting

struct

defaultValue	**float**
minValue	**float**
maxValue	**float**

5.1.5 IntProfileSetting

struct

defaultValue	**int**
minValue	**int**
maxValue	**int**

5.1.6 Profile

enum

Custom

ST20592	
G82651	
G826751	
G826752	

5.1.7 ProfileSettings

struct

logSyncInterval	**IntProfileSetting**
logAnnounceInterval	**IntProfileSetting**
logQueryInterval	**IntProfileSetting**
logMinDelayReqInterval	**IntProfileSetting**
logMinPdelayReqInterval	**IntProfileSetting**
announceReceiptTimeout	**IntProfileSetting**
syncReceiptTimeout	**IntProfileSetting**
unicastReqDuration	**IntProfileSetting**
domainNumber	**IntProfileSetting**
stepThreshold	**FloatProfileSetting**
firstStepThreshold	**FloatProfileSetting**
udpTtl	**IntProfileSetting**
dscpEvent	**IntProfileSetting**
dscpGeneral	**IntProfileSetting**
defaultProtocol	**TransportProtocol**
supportedProtocols	**list** of **TransportProtocol**
defaultDelayMechanism	**DelayMechanism**
supportedDelayMechanisms	**list** of **DelayMechanism**
defaultTransportMode	**TransportMode**
supportedTransportModes	**list** of **TransportMode**
twoStepFlag	**bool**
followUpInfo	**bool**
netSyncMonitor	**bool**
hybridMode	**bool**
inhibitMulticastMgmtMsgResp	**bool**
inhibitMgmtMsgAckForSmTlv	**bool**

5.1.8 PtpConfigDiagnostics

Provides information if system is configured in a way suitable for PTP. E.g. if IP interfaces selected for PTP are enabled, if their configuration matches PTP configuration etc.

struct

configState	**PtpSystemConfigurationState**
portsUsedForPtp	**set** of **ipTypes.IpInterfaceId**
portDiagnostics	**list** of **PtpPortDiagnosticInfo**

5.1.9 PtpCurrentData

struct

stepsRemoved	int
offsetFromMaster	bigint
meanPathDelay	bigint

5.1.10 PtpFpgaStatus

struct

locked	bool
offset	bigint
unlockedCount	bigint

5.1.11 PtpGlobalSettings

struct

twoStepFlag	bool
domainNumber	int
dscpEvent	int
dscpGeneral	int
stepThreshold	float
firstStepThreshold	float
profile	Profile

5.1.12 PtpGmData

struct

priority1	int
clockClass	int
clockAccuracy	int
priority2	int
clockIdentity	string
timeSource	string

5.1.13 PtpPortDiagnosticInfo

struct

interface	ipTypes.IpInterfaceId
diagnosticMessage	string

5.1.14 PtpPortSettings

struct

logQueryInterval	int
logSyncInterval	int
logAnnounceInterval	int
logMinDelayReqInterval	int
logMinPdelayReqInterval	int
announceReceiptTimeout	int
syncReceiptTimeout	int
udpTtl	int
delayMechanism	DelayMechanism
transportProtocol	TransportProtocol
endpoints	list of Endpoint

5.1.15 PtpPortStatus

struct

interface	ipTypes.IpInterfaceId
portState	string
peerMeanPathDelay	bigint
txStats	optional PtpStats
rxStats	optional PtpStats

5.1.16 PtpRuntimeSettings

struct

followUpInfo	bool
transportMode	TransportMode
netSyncMonitor	bool
unicastReqDuration	int
hybridMode	bool
inhibitMulticastMgmtMsgResp	bool
inhibitMgmtMsgAckForSmTlv	bool

5.1.17 PtpSettings

struct

globalSettings	PtpGlobalSettings
runTimeSettings	PtpRuntimeSettings
ports	list of ipTypes.IpInterfaceId
portSettings	PtpPortSettings

5.1.18 PtpStats

struct

sync	bigint
delayReq	bigint

pdelayReq	bigint
pdelayResp	bigint
followUp	bigint
delayResp	bigint
pdelayRespFollowUp	bigint
announce	bigint
signaling	bigint
management	bigint

5.1.19 PtpStatus

struct

currentData	PtpCurrentData
grandMasterData	PtpGmData
portData	map from string to PtpPortStatus
twoStepFlag	int
slaveOnly	bool
numberPorts	int
domainNumber	int
parentPortIdentity	string
logMinDelayReqInterval	int
logAnnounceInterval	int
announceReceiptTimeout	int
logSyncInterval	int
delayMechanism	int
logMinPdelayReqInterval	int
versionNumber	int
masterOffset	bigint
ingressTime	bigint
inSync	bool
fpgaStatus	optional PtpFpgaStatus

5.1.20 PtpSystemConfigurationState

enum

Unknown
AllCorrect
Degraded
Invalid

5.1.21 TransportMode

enum

Unicast
Multicast

5.1.22 TransportProtocol

enum

```
IPv4
IPv6
L2
```

6 systemTimeSettings (1.5)

6.1 Overview

Overview

This module contains JSON RPC API which:

- deals with system wide time configuration (NTP, PTP, manual time)
- available on MMI cards
- URL: `https://[IP ADDRESS]/mmi/api/jsonrpc`
- method names: `timex:1.1/systemTimeSettings/[METHOD NAME]`

Changelog

1.5

- moved all NTP related types and functions to a new module `ntpTypes.1.0`
- moved all PTP related types and functions to a new module `ptpTypes.1.0`
- changed PTP port ID from linux name (string) to Appear name (CTRL, D1, D2 etc.)
  - new type `ipTypes.IpInterfaceId`
  - struct `PtpSettings`, field `ports` has type changed from list of strings to list of `ipTypes.IpInterfaceId`

1.4

Added

- added support for specifying multiple interfaces to be used by PTP
  - type `PtpSettings` - field `port` renamed to `ports` and now is a list of ports to which apply given settings
  - type `PtpStatus`
    - several fields related to an interface got moved from `PtpStatus` to a new type `PtpPortStatus`
    - a new field `portData` which is a collection of `PtpPortStatus` per interface
    - renamed field `portIdentity` to `parentPortIdentity`
  - type `PtpActivePort` - as part of `TimeStatusExternal` which shows which port and slot is used to sync linuxptp if there is sync

Changed

- providing status for PTP from both MMI cards if possible (slot 1 and 2), not only from "active" MMI card
  - response type of RPC method `GetSystemTimeStatus`, i.e. type `TimeStatus` has been replaced with `TimeStatusExternal` which has PTP status fields for both MMI cards
  - type `TimeStatus` is still present because it's used internally but is not part of any RPC method any more

6.2 Command Reference

6.2.1 GetSystemTimeSettings

Returns current time settings.

- **message GetSystemTimeSettings.Request**
- **message GetSystemTimeSettings.Response**

6.2.2 SetSystemTimeSettings

Changes current time setttings.

- **message SetSystemTimeSettings.Request**
- **message SetSystemTimeSettings.Response**

6.2.3 GetCurrentUtcTime

Returns current system UTC time in seconds since beginning of unix epoch.

- **message GetCurrentUtcTime.Request**
- **message GetCurrentUtcTime.Response**

6.2.4 SetCurrentUtcTime

Changes current system time, but only if system time setting's 'timeSource' field is set to 'manual'. Otherwise this method just reports failure.

- **message SetCurrentUtcTime.Request**
- **message SetCurrentUtcTime.Response**

6.2.5 GetSystemTimeStatus

Returns data for PTP and/or NTP

- **message GetSystemTimeStatus.Request**
- **message GetSystemTimeStatus.Response**

6.3 Type Reference

6.3.1 GetCurrentUtcTime.Request

empty **struct**

6.3.2 GetCurrentUtcTime.Response

struct

`unixTimestamp`	**bigint**

6.3.3 GetSystemTimeSettings.Request

empty **struct**

6.3.4 GetSystemTimeSettings.Response

SystemTimeSettings

6.3.5 GetSystemTimeStatus.Request

empty **struct**

6.3.6 GetSystemTimeStatus.Response

struct

timeStatus	TimeStatusExternal

6.3.7 HardwareClockSettings

optional ptpTypes.PtpSettings

6.3.8 PtpActivePort

struct

slot	int
port	ipTypes.IpInterfaceId

6.3.9 SetCurrentUtcTime.Request

struct

unixTimestamp	bigint

6.3.10 SetCurrentUtcTime.Response

empty **struct**

6.3.11 SetSystemTimeSettings.Request

SystemTimeSettings

6.3.12 SetSystemTimeSettings.Response

empty **struct**

6.3.13 SystemClockSettings

SystemClockSource

6.3.14 SystemClockSource

variant

manual	SystemClockSource.manual
ntp	ntpTypes.NtpSettings
ptp	SystemClockSource.ptp

6.3.15 SystemClockSource.manual

empty **struct**

6.3.16 SystemClockSource.ptp

empty **struct**

6.3.17 SystemTimeSettings

struct

`systemclocksettings`	**SystemClockSettings** Mechanism used to control system time. 'manual' to set the system time manually. 'ntp' to set the system clock controlled by NTP daemon using provided NTP server's as the system time source. 'ptp' to set the system clock controlled by PTP using provided settings in hardwaretimesettings.
`hardwareclocksettings`	**HardwareClockSettings** Mechanism used to control hardware clock and possibly system clock. HardwareClockSettings to set hardware clock (and system clock if ptp is selected in systemclocksettings) controlled by ptp.

6.3.18 TimeStatus

DEPRECATED in RPC, for internal use only.

struct

`ntpStatus`	**optional ntpTypes.NtpStatus**
`ptpStatus`	**optional ptpTypes.PtpStatus**

6.3.19 TimeStatusExternal

struct

`ntpStatus`	**optional ntpTypes.NtpStatus**
`ptpStatus1`	**optional ptpTypes.PtpStatus**
`ptpStatus2`	**optional ptpTypes.PtpStatus**
`ptpActivePort`	**optional PtpActivePort**