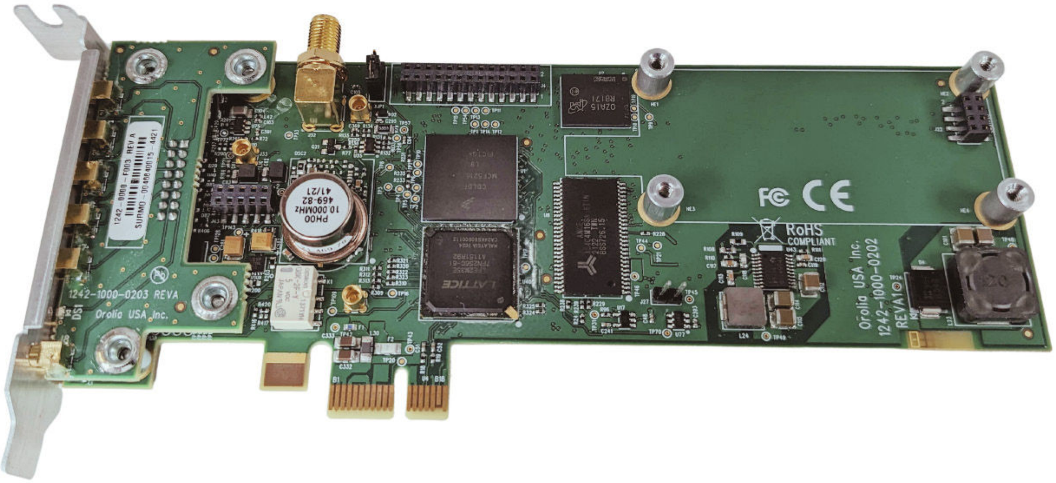
ELECTRONICS & DEFENSE

CDM-7

User Manual



1. Introduction

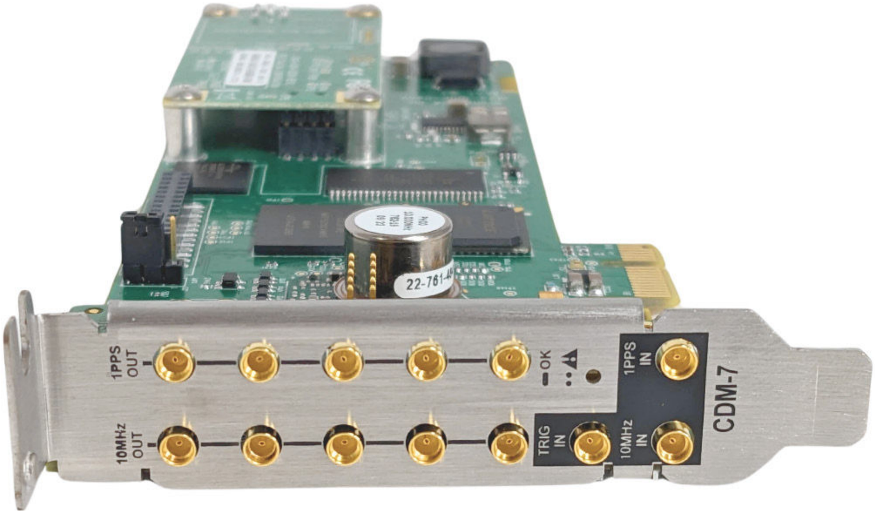
The CDM-7 clock distribution and timing reference module is a PCIe card that provides 10MHz and 1PPS signals for up to seven devices that need precise synchronization.

The CDM-7 is ideal for PCIe-based Software-Defined Radios (SDR) installed in rack-mounted or desktop PCs, and can also be used for any other application that requires a precise timing reference.

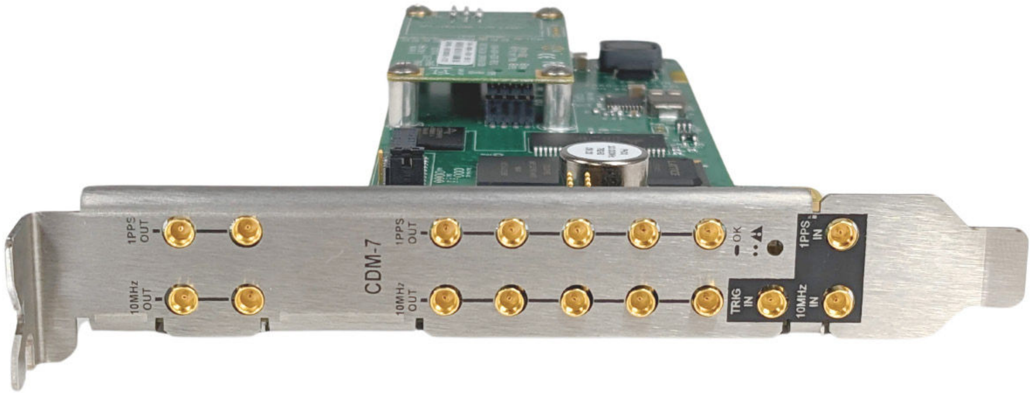
Each card has an OCXO that can phase lock to a variety of external timing references, including multi-constellation GNSS.

1.1. Options

The CDM-7 is available in two configurations. The full-height option offers 7-way distribution of 10MHz and 1PPS signals, and the half-height offers 5-way signal distribution.



CDM-7 Half-Height Model (1242-0001-0600)



CDM-7 Full-Height Model (1242-0002-0600)

Model	10 MHz IN	PPS IN	TRIG IN	10 MHz OUT	PPS OUT
Half-Height	1	1	1	5	5
Full-Height	1	1	1	7	7

2. Specifications

2.1. Timing

2.1.1. Internal Oscillator

Disciplined on-board clock

Accuracy to UTC (locked to GNSS @ 1 sigma): 325 ns

Holdover (constant temp after 2 weeks GNSS lock):

After 4 hours: 1 µs

After 24 hours: 25 µs

2.1.2. Internal Timekeeping

Timestamper Frequency: 200 MHz

Timestamper Resolution: 5 ns

External Signal Offset Measurement Resolution: 2.5 ns

2.2. Inputs

2.2.1. 1PPS

Connector: MMXC

Input Range: 2.5V…5V

Amplitude: 0 V to +5.5V, +0.8 V VIL, +2.0 V VIH

Pulse: 1Hz rising edge OR

Minimum pulse width: 100 ns

Input Impedance: <150 pF capacitive

2.2.2. 10 MHz

Connector Type: MMXC

Input Range: 0.5…5 V

2.3. Outputs

2.3.1. 1PPS

Connector: MMXC

Output Range: 5 V

Output waveform: Logic-level pulse

Time offset between any two 1PPS outputs: <50 ps

Signal level: TTL compatible, 4.3 V minimum, base-to-peak into 50 (TTL compatible, 2.2 V min, base-to-peak into high impedance)

Pulse width: Configurable pulse width (200 ms by default)

Rise time: <10 ns

2.3.2. 10 MHz

Connector: MMXC
Output Range: 2.5 V; into 50 ohm
Output waveform: Square wave
Duty cycle: 50%
Frequency accuracy: <100 ppb
Recommended warm-up time: 30 min
Minimum operational warm-up time: 5 min
Phase noise: -113 dBc @ 10 Hz, -120 dBc @ 100 Hz, -140 dBc @ 1 kHz
Harmonics: <-40 dBc
Spurious: <-70 dBc

2.4. Hardware

2.4.1. Power Supply

DC input (PCIe slot or external connector): 12 V
Current consumption: < 1 A

2.4.2. Physical

Dimensions: PCIe Standard (6.60" long)
Temperature Range: -40° C to +80° C non-condensing @ 12,000 m
Humidity: Operating and storage: 95% RH at 60° C for 5 cycles of 48 hours/cycle
Weight: PCIe: 4.3 oz/122 g
PCIe: Full-height mounting bracket provided; bus interface: low-profile PCIe x1, Rev 1.1

2.4.3. Safety & EMI

Certifications: RoHS, CE, FCC Class A

3. Operating Modes

The 6 total operating modes within the CDM-7 can be changed using API commands (see next section). The 6 modes are:

	TRIGGER MODE	TRIG_IN	10 MHz IN	PPS IN	10 MHz OUT	PPS OUT
A	TRIG SYNC	Yes	Yes	Yes	Delayed 10 MHz IN	-No signal before TRIG -Delayed PPS IN
B	TRIG ASYNC	Yes	Yes	No	Delayed 10 MHz IN	-No Signal before TRIG -<150ns after TRIG -From 10 MHz IN equates to 1PPS OUT is synchronous of TRIG IN
C	TRIG ASYNC INTERNAL	Yes	No	No	From OCXO	No Signal before TRIG -<150ns after TRIG -From OCXO
D	NO TRIG SYNC	No	Yes	Yes	Delayed 10 MHz IN	Delayed PPS IN
E	NO TRIG ASYNC	No	Yes	No	Delayed 10 MHz IN	From 10MHz in 1PPS out totally asynchrone
F	NO TRIG INTERNAL	No	No	No	From OCXO	From OCXO

4. API Commands

4.1.API Basics

> **Note:** The CDM-7 API is based on the TSync API calls. Some commands currently have responses that may need to be reinterpreted using this guide.

The CDM-7 API supports multiple devices. Most commands begin with the `<device index>` argument, which specifies which device to interact with (numbering starts from 0). If you only have one CDM-7 in your system, the `<device index>` will be 0.

The second argument to certain commands is `<index>`, which refers to multiple instances of certain components. However, in the case of the CDM-7, `<index>` will always be 0.

4.2. Managing the GNSS Reference

4.2.1. List the current reference table:

RS_GetTable

Usage: `RS_GetTable <device index> <table type>`

Valid values for <table type>:

- 0: The default table
- 1: Not applicable to CDM-7
- 2: The currently active table

Example output:

```
RS_GetTable 0 2

Idx | En  | Pri | Time | 1PPS
------------------------------
0   | DIS | 1   | gps0 | gps0
1   | DIS | 0   |      |
2   | DIS | 0   |      |
3   | DIS | 0   |      |
4   | DIS | 0   |      |
5   | DIS | 0   |      |
6   | DIS | 0   |      |
7   | DIS | 0   |      |
8   | DIS | 0   |      |
9   | DIS | 0   |      |
10  | DIS | 0   |      |
11  | DIS | 0   |      |
12  | DIS | 0   |      |
13  | DIS | 0   |      |
14  | DIS | 0   |      |
```

4.2.2. Get status of the first reference in the priority table:

RS_GetEntry

Usage: `RS_GetEntry <device index> <table entry index>`

Example output:

```
RS_GetEntry 0 0

En   | Pri | Time | 1PPS
------------------------------
DIS | 1   | gps0 | gps0
```

4.2.3. Enable the first reference (gps0):

RS_SetEnable

Usage: `RS_SetEnable <device index> <index> <enable>`

Valid values for `<enable>`:

- 0: Disable
- 1: Enable

Example output:

```
RS_SetEnable 0 0 1

  RS Table Entry 0 Enabled
```

4.2.4. Check receiver antenna status:

GR_GetAntenna

Usage: `GR_GetAntenna <device index> <index>`

Example output:

```
GR_GetAntenna 0 0

  GR (0) Antenna Status: (0) OK
```

4.2.5. Get receiver fix status (including number of satellites):

GR_GetFixData

Usage: `GR_GetFixData <device index> <index>`

Example output:

```
GR_GetFixData 0 0

  GR (0) Fix Data:
  nSats: 15
  pdop:  1.32
  hdop:  0.80
  vdop:  1.06
  tdop:  0.75
  fom:   1
  tfom:  5
  herr:  0
  verr:  1
```

4.2.6. Get receiver survey progress:

GR_GetSurveyProg

Usage: `GR_GetSurveyProg <device index> <index>`

Example output:

```
GR_GetSurveyProg 0 0

```

```
  GR (0) Survey Progress: 51%
```

4.2.7. Get current oscillator stats:

XO_GetState

Usage: `XO_GetState <device index>`

Example output:

```
XO_GetState 0

  XO State: 4
```

The numbers received at output correspond to the following states:

1. warm up
2. calibration
3. tracking setup
4. lock
5. free run

4.2.8. Get oscillator phase error:

XO_GetPhaseError

Usage: `XO_GetPhaseError <device index>`

Example output:

```
XO_GetPhaseError 0

  XO Phase Error: -57078 ns
```

4.3. Managing CDM7

4.3.1. Set the trigger mode

IR_SetMode

Usage: `IR_SetMode <device index> <index> <mode>`

Example output:

```
IR_SetMode 0 0 0

  IR (0) Rcvr Mode: (0) Auto Detection
```

You should interpret the numbers received in the output in the following manner:

0.	Auto detection	Synchronous mode (mode A)
1.	Manual	Asynchronous mode (mode B)
2.	Manual	Asynchronous mode (mode C)l
3.	Manual	Asynchronous mode (mode D)
4.	Manual	Asynchronous mode (mode E)
5.	Manual	Internal clock (mode F)
6.	Manual	Unknownl

Using LS_GetMessage to read the CDM7 log would show:

```
LS_GetMessage 0

  Log:   2000 002 21:35:49 000 CDM Mode set to A (0) for instance 0
```

4.3.2. Read the current offset

IR_GetOffset

Usage: `IR_GetOffset <device index> <index>`

Example output:

```
IR_GetOffset 0 0

  IR (0) Offset: 12855
```

Note: the value reported must be divided by 2 for nanoseconds

Using LS_GetMessage to read the CDM7 log would show:

```
LS_GetMessage 0

  Log:   2000 002 21:34:03 000 CDM Offset : 12855 (6427.5ns)
```

4.3.3. Read the current status/mode

PTR_GetMode

Usage: `PTR_GetMode <device index> <index>`

The returned value indicates the combined status and mode as a bit-field. The bits have the following meaning:

0. Fault detected
1. 10MHZ_EXT present
2. 10MHZ_OUT present
3. 1PPS_EXT present
4. 1PPS_OUT present
5. Mode A selected
6. Mode B selected
7. Mode C selected
8. Mode D selected
9. Mode E selected
10. Mode F selected

11-15 Reserved (unused)

Example output:

```
PTR_GetMode 0 0

  PTR (0) Mode: 1046
```

In the above example, the value 1046 would decode as:

No Fault
10MHz In TRUE
10MHz Out TRUE
PPS In FALSE
PPS Out TRUE
Mode F

Using LS_GetMessage to read the CDM7 log would show:

```
LS_GetMessage 0

  Log:   2000 002 21:31:51 000 CDM Status : Mode-F PPS out+in 10 MHZ out+in+
```

5. Calibration Procedure

If this unit is field installed, it is recommended to perform this calibration procedure to properly align all features and systems.

Calibration is performed by synching the system to a GNSS reference and using it to discipline the oscillator. GNSS reference setup requires manual configuration. Calibration parameters (DAC value) are stored automatically to non-volatile memory.

5.1. Calibrating multiple devices

The CDM7 API supports multiple devices. The first argument to most API calls is the device index, which specifies which device to interact with. Numbering starts from 0.

All example commands in this document use a device index of 0, indicating the first device installed in the system. To calibrate a different device, adjust the device index value accordingly.

5.2. Setup

For more detail on any of the commands listed in this section, refer to the list of API Commands.

1) Setup GNSS reference
    - a) Attach GNSS antenna to device to be calibrated
    - b) Verify antenna is connected correctly and is recognized by the system
        - i) Example command and expected result:

```
GR_GetAntenna 0 0

  GR (0) Antenna Status: (0) OK
```

    - c) Enable the GNSS reference
        - i) Example command and expected result:

```
RS_SetEnable 0 0 1

  RS Table Entry 0 Enabled
```

    - d) Set the GNSS receiver dynamics mode
        - i) Example command and expected result:

```
GR_SetMode 0 0 1 0

  GR (0) Rcvr Mode: STANDARD (1) Dynamics: (0)
```

    - e) Wait for GNSS survey to complete (indicated by progress 100%).
      Note: the survey may take up to 33 minutes to complete. If the survey progress remains at 99% after this time, it usually indicates that the receiver does not have a good fix.
        - i) Example command and expected result:

```
GR_GetSurveyProg 0 0

```

```
  GR (0) Survey Progress: 100%
```

f) Wait for GNSS reference to become valid (indicated by Time and 1PPS both returning `1`).

i) Example command and expected result:

```
GR_GetValidity 0 0

GR (0) Reference Validity:
  Time: 1
  1PPS: 1
```

2) Enable oscillator calibration.
Note: This may take a few hours to complete, depending on conditions.

a) Calibration start command:
Note: the reported mode will quickly return to the default disciplining mode `0`, as reported by `XO_GetMode`.

i) Example command and expected result:

```
XO_SetMode 0 3

  XO Mode: 3
```

b) Monitor oscillator state. The state value should proceed in order from `1` to `4`, indicating a good lock on the input reference.

i) Example command and expected result:

```
XO_GetState 0

  XO State: 4
```

c) Monitor oscillator phase error. Wait for the error to read between -200 and 200 ns.

i) Example command and expected result:

```
XO_PhaseError 0

  XO Phase Error: -57078 ns
```

3) Once the oscillator phase error is within 200 ns offset, the calibration is complete. The DAC value will be written automatically to non-volatile memory and will be used to steer the oscillator. Any of the following methods may be used to end calibration:

a) Disable the GNSS reference

i) Example command and expected result:

```
RS_SetEnable 0 0 0

  RS Table Entry 0 Disabled
```

b) Put the oscillator into freerun.

i) Example command and expected result:

```
XO_SetMode 0 1

  XO Mode: 1
```

c) Power cycle the CDM7. Upon restarting the card, the default configuration will be restored in which the GNSS reference is disabled.

4) Disconnect the GNSS antenna (optional).

6. Upgrade Procedure

Safran will occasionally issue upgrades to your device as needed. To install upgrades. you will need to use the `US_Upgrade` command.

1. Safran will provide 2 files for upgrade purposes:
   a) rt_fw.bin
   b) rt_fpga.bin

   These files will need to be uploaded to your Linux system before upgrade.

2. Run the `US_Upgrade` command separately for each upgrade file, pointing to the filepath of the location of the files on your system.

US_Upgrade

Usage: `US_Upgrade <device index> <filepath>`

Example output:

```
US_Upgarde 0 cdm7_upgrade_files/rt_fw.bin

Image Info:
 Mark:     0XDEADBEEF
 Type:     0X00000000
 Length:   0X000F4750
 Version:  49.46.48.48
 CRC:      0XD2773600
```

7. Safran Technical Support

For technical support, product specifications, and additional documentation, you can visit https://safran-navigation-timing.com/support-hub/skydel/ to submit a support request.

More information on standard unit behavior or any other features or functions of the GSG series can be found on our website at https://safran-navigation-timing.com/manuals/

Information furnished by Safran is believed to be accurate and reliable. However, no responsibility is assumed by Safran for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Safran reserves the right to make changes without further notice to any products herein. Safran makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Safran assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. No license is granted by implication or otherwise under any patent or patent rights of Safran. Trademarks and registered trademarks are the property of their respective owners. Safran products are not intended for any application in which the failure of the Safran product could create a situation where personal injury or death may occur. Should Buyer purchase or use Safran products for any such unintended or unauthorized application, Buyer shall indemnify and hold Safran and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable legal fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Safran was negligent regarding the design or manufacture of the part.

Safran Electronics & Defense
safran-navigation-timing.com