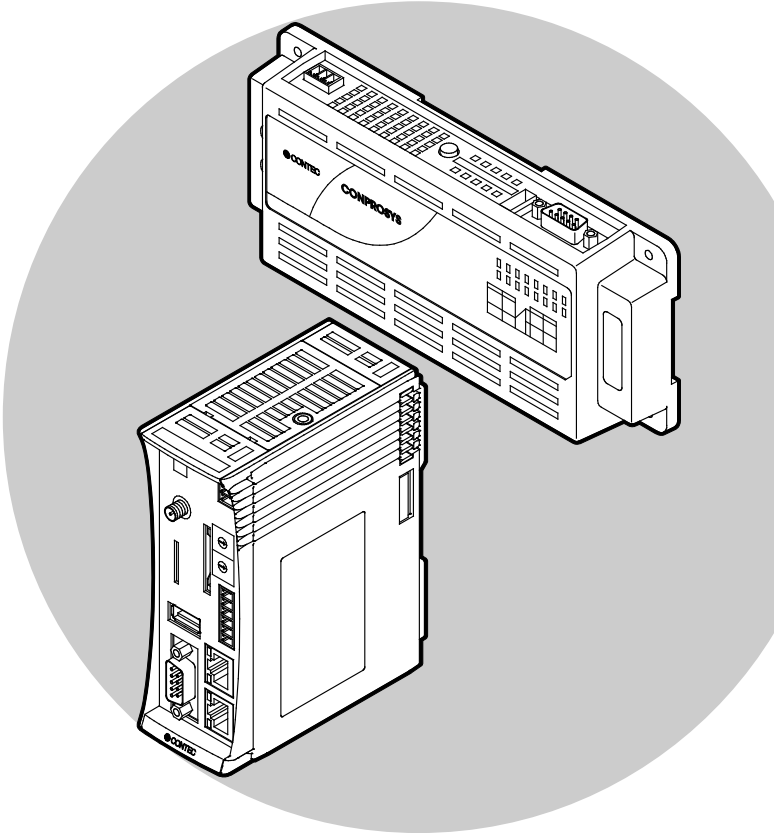
CONTEC

CONPROSYS

User's Manual

(Cross-Build Edition)

CONPROSYS Linux SDK Ver. 1.5.0



CONTENTS

CONTEC CO., LTD.

Table of Contents

Table of Contents

Introduction

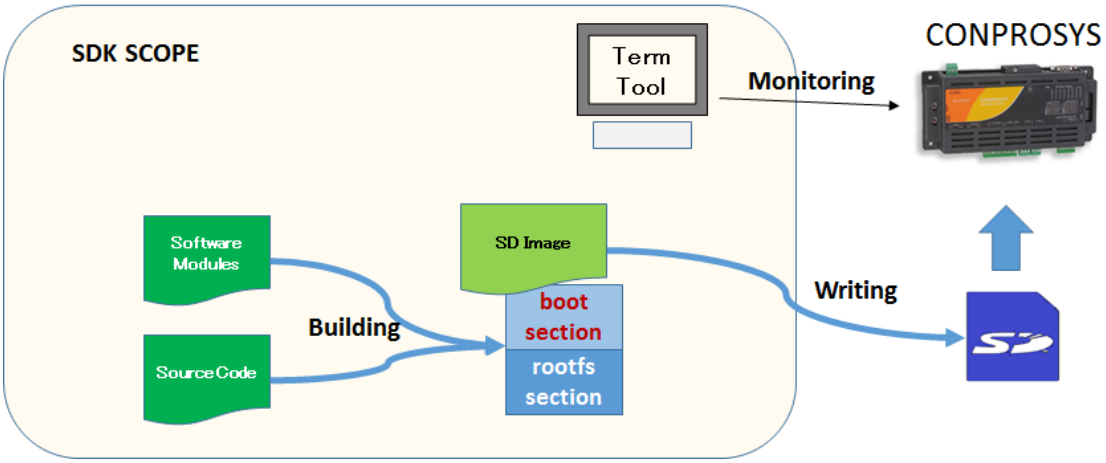
1.Outline

CONPROSYS Linux SDK (Software Development Kit) is for creating software development environment to generate modules that operate on the CONPROSYS.

SDK scope is as follows:

- Tool for generating software that CONPROSYS operates on the host PC for development. (such as source code (kernel, library, driver, etc.) and building script)
- Tool for writing CONPROSYS software on a SD card on the host PC for development.
- Tool for monitoring the software performances on CONPROSYS (such as serial console)

SDK SCOPE



This SDK generates software modules in the cross-build development on the Host PC.

If you wish to develop the software in the self-build development on CONPROSYS, please refer to the "Self-build" manual.

The SDK could also generate CONPROSYS Linux SDK in the self-build development.

Please see "**Building (page 62)**" for details.

2.CONPROSYS products

List of CONPROSYS products that support SDK

M2M Controller Series Integrated type

CPS-MC341-ADSCx series	Multi Input and Output Model
CPS-MC341G-ADSC1 series	Multi Input and Output + 3G WAN (Japan domestic/Global) Model
CPS-MC341Q-ADSC1	Multi Input and Output + 920HMz LAN (Japan only) Model
CPS-MC341-A1 series	Analog Input and Output Model
CPS-MC341-DSx series	Digital Input and Output Model
CPS-MC341-DS11	Digital Input and Output Model

M2M Gateway Series Integrated type

CPS-MG341-ADSC1 series	Multi Input and Output Model
CPS-MG341G-ADSC1 series	Multi Input and Output + 3G WAN (Japan only) Model
CPS-MG341G5-ADSC1	Multi Input and Output + LTE Model

M2M Controller Series Configurable type

CPS-MCS341-DS1 series	Controller
CPS-MCS341G-DS1	Controller+ 3G WAN (Japan only) Model
CPS-MCS341G5-DS1	Controller+ LTE Model
CPS-MCS341Q-DS1	Controller+ 920HMz LAN (Japan only) Model

M2M Gateway Series Configurable type

CPS-MGS341-DS1 series	Controller
CPS-MGS341G5-DS1	Controller+ LTE Model

* The functions such as HMI, VTC, OPC-UA, and Modbus that are installed in the M2M controller series are not provided in the CONPROSYS Linux SDK. The additional software is necessary for these functions.
* The functions such as HM, VTC, OPC-UA, Modbus, PLC, and CNC that are installed in the M2M Gateway series are not provided in the CONPROSYS Linux SDK. The additional software is necessary for these functions.
* The PAC system series and the nano series do not support the CONPROSYS Linux SDK.

3.CONTEC Software License

SOFTWARE LICENSE AGREEMENT

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT (the "Agreement") CAREFULLY BEFORE OPENING THE PACKAGE OF, DOWNLOADING, INSTALLING, OR USING THE CONTEC'S SOFTWARE. THIS AGREEMENT SET FORTH TERMS AND CONDITIONS REGARDING THE LICENSE TO USE CONTEC'S SOFTWARE DESCRIBED IN THE LICENSE SHEET (the "Software"). BY OPENING THE PACKAGE, DOWNLOADING, INSTALLING OR USING THE SOFTWARE, CUSTOMERS ARE AGREEING TO BE BOUND BY THE AGREEMENT. IF CUSTOMERS DO NOT AGREE TO THE AGREEMENT, CUSTOMERS MAY RETURN THE PRODUCT, WITHOUT OPENING THE PACKAGE, TO THE PLACE WHETERE CUSTOMERS OBTAINED IT FOR A REFUND. CUSTOMERS MAY NOT OPEN THE PACKAGE OF, DOWNLOAD, INSTALL OR USE THE SOFTWARE WITHOUT AGREEING TO THE AGREEMENT.

Article 1. Intellectual Property Rights

The copyright, patent right or any other intellectual property right pertaining to the Software or any documentary attachments, such as manuals or license sheets, as well as any copies thereof (the "Software and the Like") shall belong to CONTEC, and customers shall have no rights therefor other than those expressly authorized herein.

Article 2. Permitted License

1. CONTEC grants customers a non-exclusive right to install and use, free of charge, the Software solely for the purpose of using hardware products onto which the Software has been installed.
2. Customers may copy the Software for a minimum number of times as necessary solely for emergency backup purposes in using the Software. However, description regarding any of the rights pertaining to the Software supplied by CONTEC shall be attached to any such copies.
3. Customers may incorporate software provided by CONTEC as a library onto software created by the customer.

Article 3. Restrictions on Use

Customers shall not:

(1) Create any derivative software from the Software other than as set forth herein;

(2) Copy the Software other than as set forth therein;

(3) Modify, adapt, decompile, disassemble or reverse-engineer the Software; or

(4) Delete or alter the representation or trademark of the rights of the Software.

Article 4. Limited Liabilities

1. Except expressly warranted in article 7, CONTEC hereby disclaims any warranty with respect to the software, either express, implied or statutory, including but not limited to, the implied warranties of merchantability, of satisfactory quality, or fitness for a particular purpose and/or non-infringement of third party rights.
2. To the extent not prohibited by law, in no event shall CONTEC be liable for personal injury, or any incidental, special, indirect or any other commercial damages or losses, arising out of or related to your use or inability to use the software and the like.

Article 5. Transfer

1. Customers may transfer their rights authorized in accordance with this Agreement, to a third party only when the customer satisfies all of the following conditions:
   (1) The customer transfers the Software and the Like, or the Software installed in the hardware to such third party together with the Agreement and the license sheet; and
   (2) The recipient of the transfer agrees to the terms and conditions herein.

Article 6. Termination

1. CONTEC may terminate the Agreement with immediate effect without any notice or demand to the customer if the customer fails to comply with any of the provisions herein.
2. Upon the termination of the Agreement, the customer's license shall cease to be effective. The customer shall immediately discontinue using the Software in any way, and shall uninstall and destruct any reproduction of the Software.

Article 7. Limited Warranty on Media

1. CONTEC warrants the media on which the Software is recorded and delivered by CONTEC to be free from defects in materials and workmanship under normal use for a period of 30 days from the original retail purchase. Customers exclusive remedy shall be replacement of the Software which is returned to CONTEC through the distributor.

Article 8. Information regarding the Software

1. The information and/or updated program will be delivered through the website of CONTEC.
2. The information and/or updated program will be licensed to the Customers in accordance with the Agreement. Customers may use the information and/or updated program at their discretions and responsibilities, however, in case the Customers use such information and/or updated program they shall comply with the terms and conditions of the Agreement.

Article 9. Export Control

1. Customers shall comply with the Foreign Exchange and Foreign Trade Act of Japan, the U.S. Export Administration Regulation and the laws and regulations of any other country when taking the Software and the Like outside Japan.
2. Customers shall not transfer, export or re-export the Software and the Like to any individual or entity that is likely to use the Software and the Like to design, develop or manufacture nuclear weapons, biochemical weapons, or to design, develop or manufacture missiles.
3. Customers shall not transfer, export or re-export the Software and the Like to any individuals or entities set forth in the following countries or regions:
   (1) The Republic of Cuba, The Islamic Republic of Iran, the Republic of Iraq, the Great Socialist People's Libyan Arab Jamahiriya or North Korea;
   (2) Any individuals or entities on the "List of Foreign Users" based on the Import Trade Control Order or the U.S. Department of Commerce Denied Person's List or Entity List; or
   (3) Any country, region, individual or entity designated by the government of Japan, the U.S. or any other relevant country.

Article 10. Governing Law

The provisions herein shall be construed and governed in accordance with the laws of Japan. This Agreement shall not be governed by the United Nations Convention on Contracts for the International Sale of Goods, the application of which is expressly excluded.

Article 11. Dispute Resolution

Upon the occurrence of any dispute in relation to the Agreement or the Software, if any legal procedures are required, such as the filing of a petition for a lawsuit, the Osaka District Court shall have the exclusive jurisdiction over such dispute.

Article 12. Severability

If for any reason any portion of the provisions set forth herein is found to be invalid or unenforceable, the remainder of the Agreement shall not be affected in any way and shall be valid and enforceable to the extent permitted by law.

Safety Precautions

1. Safety Information

This document provides safety information using the following symbols to prevent accidents resulting in injury or death and the destruction of equipment and resources.

Understand the meanings of these labels to operate the equipment safely.

⚠ **DANGER**	Signal word used to indicate an imminently hazardous situation which, if not avoided, will result in death or serious injury.
⚠ **WARNING**	Signal word used to indicate a potentially hazardous situation which, if not avoided, could result in death or serious injury.
⚠ **CAUTION**	Signal word used to indicate a potentially hazardous situation which, if not avoided, could result in minor or moderate injury.

2. Handling Precautions

⚠ CAUTION

- The specifications of the product are subject to change without notice for enhancement and quality improvement. Even when using the product continuously, be sure to read the manual in the CONTEC's website and understand the contents.
- Do not modify the software.
  CONTEC will bear no responsibility for any problems, etc., resulting from modifying the software.
- Regardless of the foregoing statement, CONTEC assumes no responsibility for any errors that may appear in this document or for results obtained by the user as a result of using the software.

3. Security Warning

When connecting to the network, be aware of security-related problems. See the examples of Security measures below and set up the product properly along with the network devices.

1. Information security risks

- Unauthorized access from the outside through a network could cause the system halt, data damage, or exposure to malware. *1
- Invaded and used as a stepping stone, a device might attack the others through networks. (a victim becomes an assailant)
- Information might leak without realizing due to the connection to the network.
- Secondary damages such as harmful rumors, liability in damages, social credibility fall, and opportunity loss are expected led by the troubles described above.

*1: Malware (Malicious Software) is software that brings harm to a computer system and performs unintended operations.

2. Security measures – e.g.

- Do not keep using the default password. (Refer to the product manual for the password setting).
- Set a strong password.

  Combined with upper and lowercase letters, and numbers so that it cannot be easily analogized by others.

- Change the password periodically.
- Disable unnecessary network services and functions.
- Restrict access to the network with network devices. *2
- Restrict ports to be released on the network with network devices. *2
- Create a closed network connection using such as dedicated network or VPN*3

*2: Inquire for setting procedure to manufacturers.

*3: VPN (Virtual Private Network) a secured network that wards off unauthorized access by protecting the communication path with authentication and encryption.

Unfortunately, there are no perfect ways to avert unauthorized access or close a security hole that are endlessly found day and night.

Please understand that risks are always involved with the Internet connection, and we strongly recommend a user should constantly update information security measures.

Development environment

1. Required items for development

- Host PC for development (Linux)
- SDHC card (2Gbyte or larger, non-SDXC-compliant)
- Cable for serial monitor (Recommended cable: TTL-232R-3V3-AJ by FTDI)
- LAN Cable

2.SDK specification

Host PC for development Linux Distribution:	Ubuntu 14.04 / 16.04 (64bit version) Desktop 40Gbyte or larger HDD spare capacity required User who can execute sudo command with administrative privileges.
Kernel version for target:	3.2.0
Distribution for target:	arm version Ubuntu 14.04 (for SD booting only)
Cross-compile GCC version:	gcc 4.9 (Hardware float) / gcc 4.7 (Software float)

Required Linux toolchain:

apt, gcc-arm-linux-gnueabi, libncurses5-dev, gawk, u-boot-tools, openssh-server, samba, binutils-arm-linux-gnueabi, binutils-arm-linux-gnueabihf, xinetd, kpartx, gperf, bison, flex

* Above are requirements to run the SDK.
Please install other necessary packages to accommodate your development environment.
(e.g., git, wget, subversion)

3.SDK contents

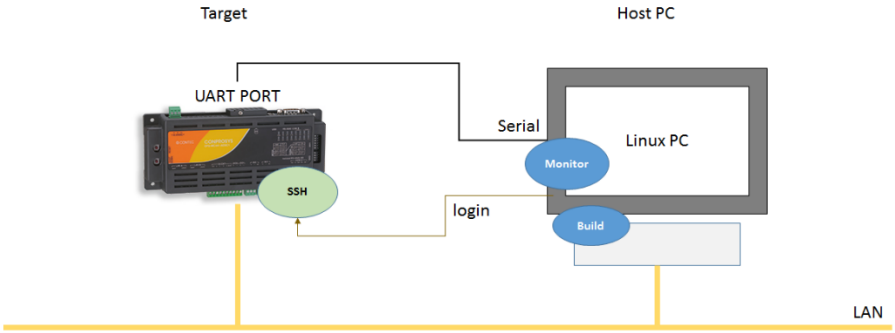
- SDK document
- Cross-compiler/Tool chain
  Package (for Debian/Ubuntu):
  gcc-arm-linux-gnueabihf-4.9.3, gcc-arm-linux-gnueabi-4.7, libncurses5-dev, gawk, u-boot-tools, openssh-server, samba, binutils-arm-linux-gnueabi, binutils-arm-linux-gnueabihf, xinetd, kpartx, gperf, bison, flex
- Build tools
- Source code
  u-boot, kernel, sample applications, sample libraries, sample drivers
- Base module per CONPROSYS product (such as u-boot, kernel, setup)

4. Development environment composition

Example compositions of the host PC (for building and monitoring) and target are shown below.

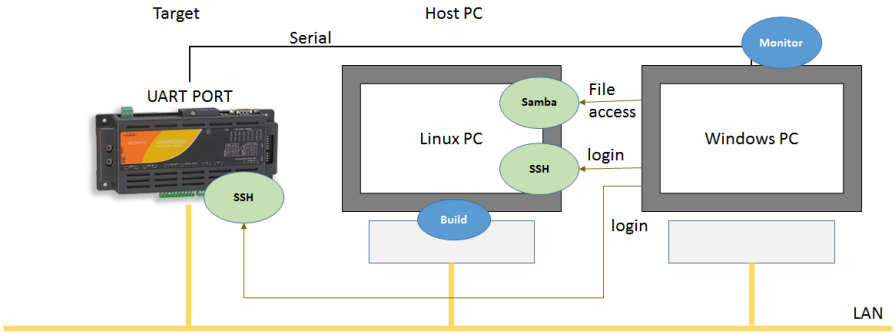
Example 1) with one development host PC to build software and use a serial monitor for target.

One Linux PC for building software and serial monitoring



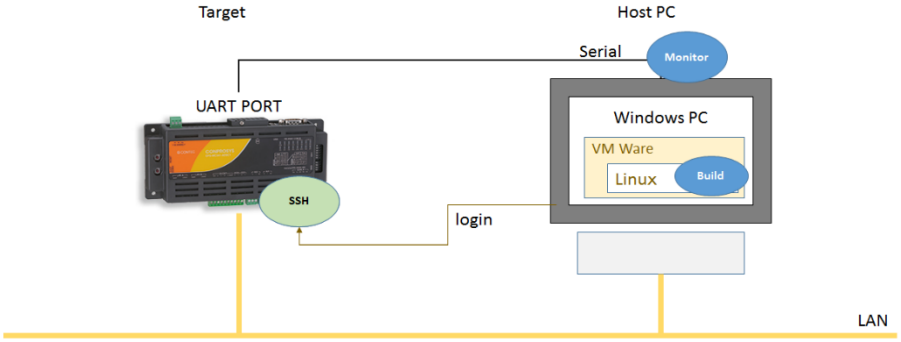
Example 2) with one development host PC to build software (or source code edit), use a serial monitor for target with other Windows PCs.

One Linux PC for building software, and Windows PCs for serial monitoring



Example 3) to use as a development host PC by installing virtual OS system (VM Ware, Virtual BOX etc.) into Windows OS, and installing Linux OS from the virtual system.

One PC to use Linux (with VM Ware) for building software, and Windows PC for serial monitoring



5. SDK installation

Installation by downloading or from DVD described below.

◆ By downloading (tgz file):

1 Extract the downloaded .tgz file.
tar xvfz CPS_SDK_installer_xxxx.tgz[–C ExtractDirectory]

2 It will be navigated to the extract directory.
*This is not required when the extract directory is current.

◆ By downloading (iso file):

1 Mount the downloaded .iso file.
Create a customized mount directory.
sudo -E mount –o loop CPSSDK_xxxx.iso MountTargetDirectory

2 It will be navigated to the mount target directory.

◆ From DVD:

1 Install DVD media into the host PC.

2 The installed media is automatically mounted and navigated to the directory.

1. Tool chain installation necessary for SDK

◆ When the host PC can be connected to the internet

Install the following tool chain on ubuntu OS.

libncurses5-dev, gawk, u-boot-tools, openssh-server, samba, binutils-arm-linux-gnueabi, binutils-arm-linux-gnueabihf, xinetd, kpartx, gcc-4.7-arm-linux-gnueabi, gperf, bison

Update the "apt-get" package list before installing tool chain.

Update command for the list:

```
sudo apt-get update
```

Installation command:

```
sudo apt-get install libncurses5-dev gawk u-boot-tools openssh-server samba ¥
  binutils-arm-linux-gnueabi binutils-arm-linux-gnueabihf xinetd kpartx gperf ¥
  bison flex
```

Install the compiler from the package of CONPROSYS linux SDK.

Installation command:

```
cd Toolchain
sudo ./compiler_pkginstall.sh
cd ..
```

◆ When the host PC cannot be connected to the internet

We are offering the packages of required tool chain in CONPROSYS linux SDK.

Please go to [Toolchain] directory and run "toolchain_pkginstall.sh".

(or you could install it through "./install_sdk.sh" as instructed in "**CONPROSYS linux SDK installation (page 22)**").

Command:

```
cd Toolchain
sudo ./toolchain_pkginstall.sh
cd .
```

2. CONPROSYS linux SDK installation

Start SDK installation with the command below.

Command:

```
./install_sdk.sh [–C InstallTargetDirectory] [-t]
```

Option:

-C Installation directory
Generate a specified installation directory and install the SDK under the directory.

-t
Install necessary tool chain such as cross-compiler for SDK.
When this option is specified, the administrator password is requested to install the tool chain into the development host PC.

* When an installation directory is not specified, a directory named "CPS_SDK" is automatically generated under the current directory, and the tool chain is installed into that directory.
A directory cannot be generated under the current directory by mounting iso file or DVD media installation. Make certain to specify the directory to install.

* For installation directory, the location under the home directory of the login user is recommended.
Command example:

```
./install_sdk.sh -C ~/CPS_SDK
```

The following diagram is the directory composition after installation.

Directory composition diagram after installation

```
~/
 |
 +-- [CPS_SDK]
        |
        +-- [Document]     Document
        |
        +-- [application]  Application source directory
        |
        +-- [base]  Default file storage directory
        |
        +-- [driver]       Driver source directory
        |
        +-- [kernel]       kernel source directory
        |
        +-- [lib]          Library source directory
        |
        +-- [ramdisk]      ramdisk (for NOR FLASH) generating directory
        |
        +-- [rootfs]       rootfs source directory
        |
        +-- [tools] Tools
        |
        +-- [u-boot]       u-boot source directory
        |
        +-- [target]       Target directory
```

[Document]

A directory to store SDK document files.

[application]

A directory to store application source code.

[base]

A directory to store boot and rootfs sections that are bases of target.

[tools]

SDK tools.

[driver]

A directory to store driver source code.

[kernel]

A directory to store kernel source code.

[lib]

A directory to store library source code.

[ramdisk]

A directory to generate ramdisk packaging rootfs that to be installed into NOR FLASH.

[rootfs]

A directory to store source code of light version rootfs (root file system) that is used for built-in NOR FLASH booting.

[u-boot]

A directory to store u-boot source code.

[target]

A directory to generate a section for starting SD per CONPROSYS product.

After executing configure.sh, a target directory is generated and this can a storage destination for built modules (boot, kernel, drivers, and applications).

Cross-Build Environment

1. Creating a SD card

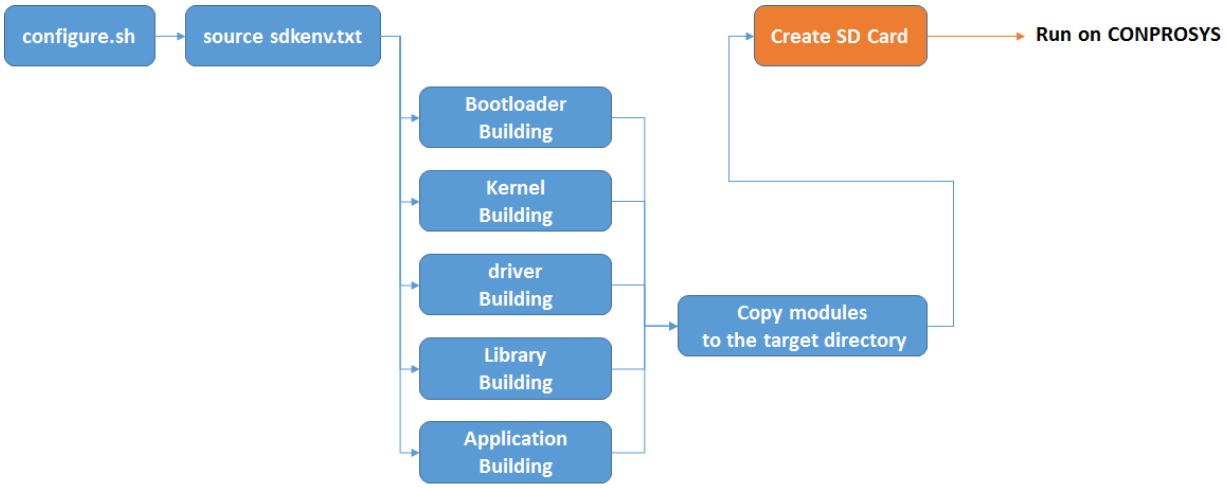
The diagrams below show the flow from generating a module to creating a booting SD card.

◆ Target in the self-build development on CONPROSYS Linux SDK

Creating a SD card of CONPROSYS Linux SDK in the self-build development

◆ Target in the SDK cross-build development

Creating a SD card in the cross-build development



Please refer to "**Create a SD card for start-up (page 33)**" for creating a SD card.

Please refer to "**Building (page 62)**" and the following pages regarding Building.

2.Initial Settings

Under the SDK installation directory, configure the initial settings to run "./configure.sh" for building. This command generates the following environment file and directory. The command is only used to generate a new module for targeting.

- Environment configuration file for building (sdkenv.txt)
- Kernel configuration file
- File system to write on a SD card that complies with each product under the target directory. ("boot" section, "rootfs" section)

Command:

```
./configure.sh
```

*Password might be requested to execute root privileges command during the setting.
Please enter the password and continue the procedure.

Execute command and the list of CONPROSYS models is displayed. Enter the corresponding number to the specified product to continue.

CONPROSYS Product:

Select a number between 1) and 8) in accordance with the target model.

1) CPS-MC341-ADSCx	Integrated type products Multi-I/O Model CPS-MC341-ADSCx Series, CPS-MC341G-ADSC1 Series (3G WAN Model), CPS-MC341Q-ADSC1(920HMz LAN Model(Japan only)), CPS-MG341-ADSC1 Series, CPS-MG341G-ADSC1 Series (3G WAN Model), CPS-MG341G5-ADSC1(LTE Model)
2) CPS-MC341-Ax	Integrated type product Analog I/O Model CPS-MC341-A1
3) CPS-MC341-DSx	Integrated type product Digital I/O Model CPS-MC341-DSx Series
4) CPS-MC341-DS1x	Integrated type product Digital I/O Model (with USB port) CPS-MC341-DS11
5) CPS-MxS341-DSx	Configurable type products CPS-MxS341-DS1 Series, CPS-MCS341G-DS1(3G WAN Model), CPS-MxS341G5-DS1(LTE Model), CPS-MCS341Q-DS1(920HMz LAN Model (Japan only))

LAN type:

Select a number in accordance with LAN type.

1) 1lan (HUB mode) (SINGLE EtherMAC)	EtherMAC is recognized as one. Ether port is used as HUB mode.
2) 2lan (DUAL EtherMAC)	Each EtherMAC is used per Ether port.

Select 2lan when connecting Ether A port directly to PC for debugging, and connecting Ether B port to internet as in the rootfs type such as CONPROSYS Linux SDK self-build.

rootfs type:

Select a number between 1) and 4) in accordance with the target rootfs type.

Type 2) and 4) with SDK contain a self-compiler and you could develop software on CONPROSYS.

1) light (busybox)	rootfs - light version
2) Ubuntu 14.04	Ubuntu 14.04
3) Ubuntu 14.04 (include SDK)	Ubuntu 14.04 (CONPROSYS Linux SDK self-build Edition)

Tool choice:

When the light version rootfs is selected as Rootfs type, tools can be loaded with your choice.

Decide the tool type and enter the corresponding number

1) Wireless tools, Apache 2.4, PHP5	Wireless tool, Apache 2.4, PHP5
0) None	No tool

Cross compiler type:

Select a number in accordance with cross compiler type when you have chosen a rootfs light version.

1) gnueabi (default)	arm cross compiler
2) gnueabihf	cross compiler which supports arm hardware float

The gnueabihf is chosen automatically when selecting the rootfs of Ubuntu 14.04.

(The selection list is not displayed)

After executing "configure.sh", target directory, a boot section that becomes a base, and directory/file of rootfs section chosen in ./configure are generated under the target directory

The diagram below is an example of directory upon specifying "CPS-MC341-ADSC series 2 LAN" type / Ubuntu14.04 with SDK.

Directory composition diagram under target

```
~/
 |
 +-- [CPS_SDK]
        |
        +-- [target]                                Target directory
              +-- [CPS-MC341-ADSCX.2lan]            for CPS-MC341-ADSCX (2 Lan type) directory
                    +-- [boot]                      boot section
                     |        +-- MLO
                     |        +-- u-boot.img
                     |        +-- uEnv.txt
                     |        +-- uImage
                    +-- [Ubuntu14.04_dev]           rootfs section (for a SD card)
                              +-- [bin]
                              +-- [boot]
                              +-- [dev]
                              +-- [etc]
                              +-- [home]
                              +-- [lib]
                              +-- [media]
                              +-- [mnt]
                              +-- [opt]
                              +-- [proc]
                              +-- [root]
                              +-- [sbin]
                              +-- [selinux]
                              +-- [srv]
                              +-- [sys]
                              +-- [tmp]
                              +-- [usr]
                              +-- [var]
```

3.Environment Settings

Before building applications or kernel, set the environment variable to build with "sdkenv.txt" generated by "./configure.sh" under the SDK installation directory.

Command:

```
source sdkenv.txt
```

If you omit this setting, operations such as building or firmware writing method described in the following

sections might not function properly.

How to write firmware for target

1. About system start-up

The following methods are available for target start-up.

◆ Starting system from the SD card

Write firmware on a SD card and insert the SD card to CONPROSYS for starting the system.

◆ Starting system from built-in NOR FLA SD SH

Follow the steps described below to start the system from built-in NOR FLASH

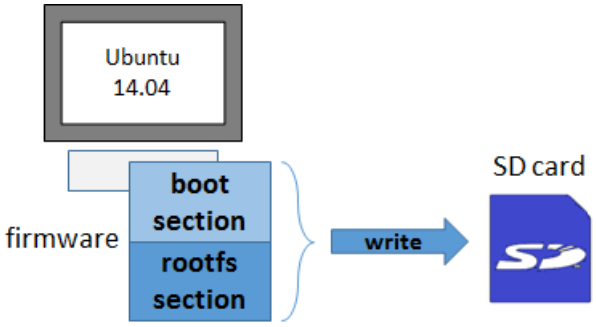
1. Write firmware and create a SD card to install the system into built-in NOR FLASH.
2. Insert the SD card to start and install the system into built-in NOR FLASH.

This section explains how to write firmware appropriate for each start-up

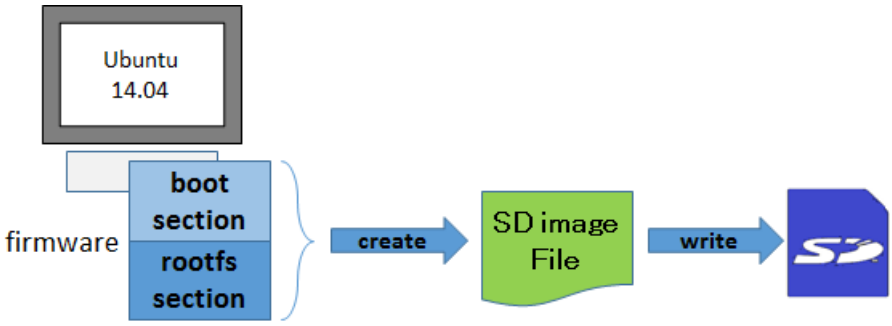
2.Create a SD card for start-up

Follow the steps described below to make a booting SD card.

1) Write firmware directly into SD card.



2) Create a SD image file and write it into the SD card by software.



1. Write firmware directly into SD card

1 Insert the SD card for the host PC to recognize it.
You can check with a parted command to see which file system the SD card is recognizing.
Example)

```
sudo parted -l
```

If the SD card is automatically mounted, please unmount it.

2 Generate SD card partitions.
Example) "/dev/sdb" for a SD card device

```
sudo -E $CPS_SDK_ROOTDIR/tools/mk2PartSDCard.sh /dev/sdb
```

This command generates two partitions listed below.

- Boot partition W95 FAT32 (LBA)
- rootfs partition ext3

3 Mount the SD card.
Create a directory beforehand to mount partitions generated in the previous section.
The followings are example commands to generate a directory for boot and rootfs under "/media".

```
sudo mkdir /media/boot
sudo mkdir /media/rootfs
```

Mount the partitions of SD card generated in the previous section to these mounting directories.
The followings are example commands in the case of "/dev/sdb".

```
sudo mount /dev/sdb1 /media/boot
sudo mount /dev/sdb2 /media/rootfs
```

4 Copy the created files of boot and rootfs directory into the SD card under the target.

[boot partition (fat32)]

```
sudo cp -p ${CPS_SDK_INSTALL_FULLDIR}/boot/* /media/boot
```

[rootfs partition (ext3)]

```
sudo -E cp -rp ${CPS_SDK_INSTALL_FULLDIR}/Ubuntu10.04/* /media/rootfs
```

5 Synchronize the file that is copied on the SD card

```
sync
sync
sync
```

If you unmount the SD card before synchronizing with the sync command and remove the SD card, a file might not be written properly. To prevent such a case, please execute the sync command first.

6 Unmount the SD card and remove it from the product.

```
sudo umount /media/boot
sudo umount /media/rootfs
```

2. Create a SD image file and write it into the SD card by software.

1 Create a SD image file

Create the file by the command below.

Command:

```
sudo -E ${CPS_SDK_ROOTDIR}/tools/mk2SDCardImage.sh ${CPS_SDK_ROOTFS} [–f filename] [–s size]
```

Option:

-f filename

The name of the image file to output can be specified. If not specified, the firmware is output into SD.img.

-s size

The size of the image file to output can be specified. If not specified, the firmware is output with the size of 2000Mbyte.

A command example: the file name is Target name _rootfs.img, and the size is 4000Mbyte

```
sudo -E ${CPS_SDK_ROOTDIR}/tools/mk2SDCardImage.sh ${CPS_SDK_ROOTFS} ¥
–f ${CPS_SDK_TARGET_NAME}_${CPS_SDK_ROOTFS}.img –s 4000
```

2 Write the image file into SD card.

[For Windows]

The following example demonstrates writing procedure with Win32 Disk Imager.

Before you start, download the installer of the Win32 Disk Imager from the website (see below) and

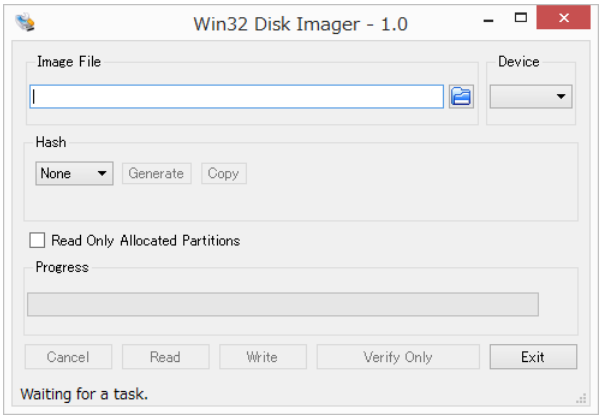
install it in the Windows PC.

https://sourceforge.net/projects/win32diskimager/

i) Insert the SD card into Windows PC.

ii) Start Win32 Disk Imager.

Win32 Disk Imager Application



iii) Select an image file to write.

Check whether SD card is selected for writing destination in the Device field.

Click "Write" button.

iv) Pop-up message appears to notify of the writing completion.

Click "Write" button and remove the SD card.

[With Linux]

i) Unmount the SD card if it is mounted.

```
sudo umount /dev/sdb
```

ii) Write the image file into SD card by dd commnad.

```
sudo dd if=image file name of=/dev/sdb bs=1M
```

iii) Synchronize the file by the command, sync.

```
sync
```

iv) Remove the SD card when the synchronizing is completed.

3. Write installing software on a SD card for built-in NOR FLASH

Before writing, the preparation described below needs to be done.

- Create a rootfs section to install built-in NOR FLASH.
- Build customized bootloader or application software and copy the results into the rootfs section for installation.

1. Create a rootfs section to install built-in NOR FLASH

Create the base rootfs (InstallerForFlash) for installing built-in NOR FLASH.

Command:

```
./create_FlashInstaller.sh
```

After executing this command, a directory "InstallerForFlash" is generated under the directory for target.

The example below shows the directory generated under the building environment of CPS-MC341-ADSC series 2 LAN type.

Composition of the directory when rootfs is generated for NOR FLASH installation

```
~/
 |
 +-- [CPS_SDK]
        |
        +-- [target]          Target directory
              +-- [CPS-MC341-ADSCX.2lan]      for CPS-MC341-ADSCX (2lan type) directory
                    +-- [boot]                boot section
                    +-- [Ubuntu14.04_dev]     rootfs section (for a SD card)
                    +-- [InstallerForFlash]   rootfs section (for NOR FLASH installation)
                          |
                          +-- [InstallToMTD] NOR FLASH installing file directory
                          |       +-- MLO.byteswap NOR Flash version bootloader
                          |       +-- u-boot.img   NOR Flash version u-boot
                          |       +-- ramdisk.xz   ramdisk
                          |       +-- uImage       kernel
                          |       +-- mtd5.tgz     application data
                          +-- [bin]
                          +-- [dev]
                          +-- [etc]
                          +-- [home]
                          +-- [lib]
                          +-- [libexec]
                          +-- linuxrc
                          +-- [mnt]
                          +-- [proc]
                          +-- [sbin]
                          +-- [sys]
                          +-- [tmp]
                          +-- [usr]
                          +-- [var]
```

Application data (mtd5.tgz)

```
[mtd5.tgz]
    |
    +-- [etc]
    |       +-- passwd
    |       +-- group
    +-- [conprosys]
    |       +-- config.ini
    |       +-- CPS_SDK_VER i
    +-- [opt]
    +-- startup.sh
    +-- CPS_SDK_VER
```

2. Copy the results into the rootfs section for installation

Copy the results from the customized bootloader or application software into installation directory (/InstallToMTD) for rootfs (InstallerForFlash).
Check whether there is a file generated by building before copying.

[bootloader]
Copy MLO.byteswap u-boot.img that was built in "**Building for built-in NOR FLASH (page 64)**".
Please see "**Building for built-in NOR FLASH (page 64)**" for the procedure.
Command:

```
cd ${CPS_SDK_ROOTDIR}/u-boot
cp -p MLO.byteswap u-boot.img ${CPS_SDK_INSTALL_FULLDIR}/InstallerForFlash/InstallToMTD
```

[kernel]
Command:

```
cd ${CPS_SDK_ROOTDIR}/kernel
cp -p arch/arm/boot/uImage ${CPS_SDK_INSTALL_FULLDIR}/InstallerForFlash/InstallToMTD
```

[ramdisk]
Command:

```
cd ${CPS_SDK_ROOTDIR}/ramdisk
make install
```

The program (shell script) contained in this SDK installs these four files (MLO.byteswap, u-boot.img, uImage, and ramdisk.xz) and the product own booting shell (startup.sh) into NOR FLASH.
Installation into "u-boot option" needs the following file editing.

${CPS_SDK_INSTALL_FULLDIR}/InstallerForFlash/home/flashwriter.sh

The content below is commented out in this file as a reference.

Copy with mtd2: dd

Please use this reference to customize your installation.
See "**Built-in NOR FLASH memory map (page 93)**" for details such as capacity of NOR FLASH.

3. Write installing software on a SD card for built-in NOR FLASH (directly on a SD card)

Follow the steps described below.

The procedure is basically the same as SD card booting, however, rootfs directory to be copied is different and it is "InstallerForFlash".

1 Insert the SD card for the host PC to recognize it.
You can check with a parted command to see which file system the SD card is recognizing.
Example)

```
sudo parted -l
```

2 Generate SD card partitions.
Example) /dev/sdb for SD card device

```
sudo -E $CPS_SDK_ROOTDIR/tools/mk2PartSDCard.sh /dev/sdb
```

This command generates two partitions listed below.

- - Boot partition W95 FAT32 (LBA)
- - rootfs partition ext3

3 Mount the SD card.
Create a directory beforehand to mount partitions generated in the previous section.
The followings are example commands to generate a directory for boot and rootfs under /media.

```
sudo mkdir /media/boot
sudo mkdir /media/rootfs
```

Mount the partitions of SD card generated in the previous section to these mounting directory.
The following are example commands in the case of /dev/sdb.

```
sudo mount /dev/sdb1 /media/boot
sudo mount /dev/sdb2 /media/rootfs
```

4 Copy the created files of boot and rootfs directory into the SD card under the target.

[boot partition (fat32)]

```
sudo cp -p ${CPS_SDK_INSTALL_FULLDIR}/boot/* /media/boot
```

[rootfs partition (ext3)]

```
sudo -E cp -rp ${CPS_SDK_INSTALL_FULLDIR}/InstallerForFlash/* /media/rootfs
```

5 Synchronize the file that is copied into the SD card

```
sync
sync
sync
```

If you unmount the SD card before synchronizing with a sync command and remove the SD card, a file might not be written properly. To prevent such a case, please execute the sync command beforehand.

6 Unmount the SD card and remove it from the product.

```
sudo umount /media/boot
sudo umount /media/rootfs
```

4. Create a SD card for built-in NOR FLASH installation (Create a SD image file)

1 Create a SD image file
Create the file by the command below.

Command:

```
sudo -E ${CPS_SDK_ROOTDIR}/tools/mk2SDCardImage.sh InstallerForFlash [–f filename] [–s size]
```

Option:

-f filename

The name of the image file to output can be specified. If not specified, the firmware is output into SD.img.

-s size

The size of the image file to output can be specified. If not specified, the firmware is output with the size of 2000Mbyte.

A command example: the file name is "Target name _InstallerForFlash.img"

```
sudo -E ${CPS_SDK_ROOTDIR}/tools/mk2SDCardImage.sh InstallerForFlash ¥
-f ${CPS_SDK_TARGET_NAME}_InstallerForFlash.img -s 256
```

2 Write the image file in the SD card.

How to write the image file into the SD card, see "**2) Create a SD image file and write it into the SD card by software. (page 33)**".

4. Installation into built-in NOR FLASH

Enable SD boot of CONPROSYS with the SD card created in "**Write installing software on a SD card for built-in NOR FLASH (directly on a SD card) (page 41)**" or "**Create a SD card for built-in NOR FLASH installation (Create a SD image file) (page 43)**", and start booting.

Writing into NOR FLASH automatically begins.

Please see "**Booting from the SD card (page 46)**" for start-up details.

LEDs of ST1 (Green), ST2 (Red), and Power (Green) keep flashing upon installation and

LEDs of ST1 (Green) as well as Power (Green) remain lit after the completion of installing.

Target operation check

1. Target booting

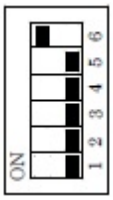
1. Booting from the SD card

Check DIP SW of each CONPROSYS product and make sure SD boot mode is enabled.

◆ Integrated type (e.g., CPS-Mx341-xxx) (Including Gateway Series)

No.6 of DIP SW1 is ON. (SD boot mode enabled)

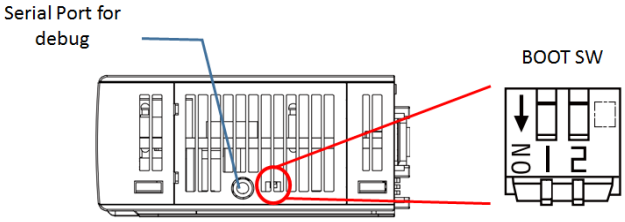
Integrated type BOOT SW setting



◆ Configurable type (e.g., CPS-MxS341-xxx)

No.2 PIN of BOOT SW (inside of the case) next to the debagging serial port (3.5Φ MINI-JACK) is ON. (SD boot enabled)

Configurable type BOOT SW setting



Insert the SD card you made in "**Create a SD card for start-up (page 33)**" and turn on the product power. Without the SD card, booting starts from built-in NOR FLASH.

2. Booting from built-in NOR FLASH

Check the SD boot mode described in "**Booting from the SD card (page 46)**" is disabled and turn on the product power.

2. Login by serial cable connecting

By connecting a serial cable from the host PC to CONPROSYS through the serial port (3.5Φ MINI-JACK), you can login from the console to the product.

See the setting details below.

Baud rate:	115200 bps
Data bit:	8 bit
Parity:	none
Stop bit:	1 bit
Hardware flow:	none

We recommend the following serial cable for connecting the host PC to CONPROSYS.

Download appropriate driver software to match OS of PC for serial monitoring.

- TTL-232R-3V3-AJ by FTDI
  URL for driver: http://www.ftdichip.com/Drivers/VCP.htm

Default login and password are

login:	conprosys
password:	contec

*When your environment is connectable to an external network, be sure to change the password as in guided in "**Security Warning (page 13)**".

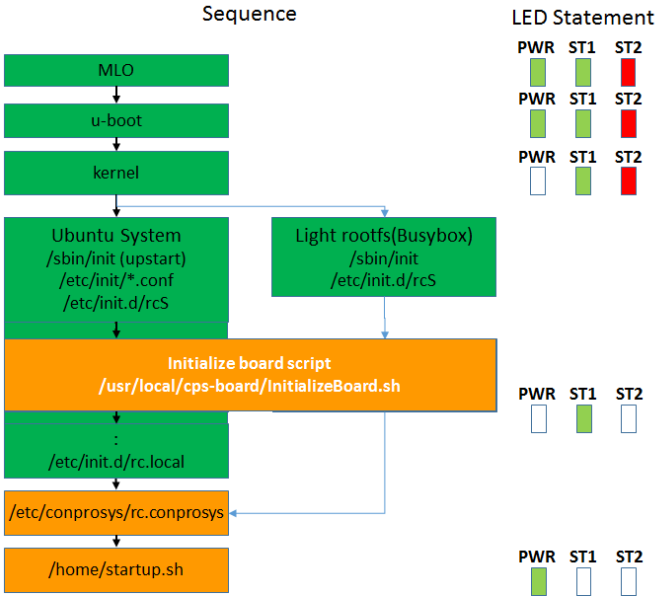
3. Login through SSH connecting

When the host PC and the CONPROSYS are operating on the same network, you can login through SSH to the product.

4. Target starting sequence

The starting sequence on target is listed below.

Starting sequence



If you would like to run commands upon booting, edit script files below.

[SD card booting]

`/home/startup.sh`

[built-in NOR FLASH booting]

`/mnt/mtd/startup.sh`

To edit the files, change the mode to "writable" since the system is set in "read only" on target. Switch back the mode to "read only" after editing.

[SD card booting]

WRITABLE mode

`sudo rommode rw`

READ ONLY mode

`sudo rommode ro`

[NOR FLASH booting]

WRITABLE mode

`sudo rommode_mtd rw`

READ ONLY mode

`sudo rommode_mtd ro`

5. Target network setting

Default network setting is listed below.

[Default setting]

LAN A (eth0): 10.1.1.101

LAN B (eth1): DHCP (for 2 LAN Type setting only)

If you would like to change the network setting, edit the file according to each root privilege.

If you would like to change the network setting, edit the file on target below with root privileges.

/etc/conprosys/config.ini

LAN Configuration

Item name	Description
eth0_dhcp	Set DHCP enabled or disabled of LAN A (eth0). enabled disabled
eth0_ipaddr	Set IP address of LAN A (eth0).
eth0_netmask	Set a netmask of LAN A (eth0).
eth0_gateway	Set a gateway address of LAN A (eth0).
eth0_dns1	Set DNS server address of LAN A (eth0).
eth1_dhcp	Set DHCP enabled or disabled of LAN B (eth1). enabled disabled
eth1_ipaddr	Set IP address of LAN B (eth1).
eth1_netmask	Set a netmask of LAN B (eth1).
eth1_gateway	Set a gateway address of LAN B (eth1).
eth1_dns1	Set DNS server address of LAN B (eth1).
ntp_addr	Set NTP server.
host_name	Set a host name. The following name is used as the default host name since there is no item in the setting. The model name + the lower 3 bytes of MAC address

3G/LTE network configuration (3G/LTE model exclusively)

Item name	Description
m3g_connect	Set 3G/LTE connection enabled or disabled. enabled disabled
m3g_apn	Set APN supplied by network service provider.
m3g_user	Set User ID supplied by network service provider.
m3g_passwd	Set password supplied by network service provider.
m3g_auth	Set the following encryption supplied by network service provider. None PAP CHAP

Wireless LAN configuration

Item name	Description
wlan_dhcp	Set DHCP enabled or disabled of wireless LAN (wlan0). enabled disabled
wlan_ipaddr	Set IP address of wireless LAN (wlan0).
wlan_netmask	Set a netmask of wireless LAN (wlan0).
wlan_gateway	Set a gateway address of wireless LAN (wlan0).
wlan_dns1	Set DNS server address of wireless LAN (wlan0).
wlan_essid	Set SSID of wireless LAN (wlan0).
wlan_encrypt	Set a cipher for wireless LAN (wlan0) by choosing one listed below [Setting items] No cipher: none WEP: wep WPA-PSK AES: wpapsk-aes WPA-PSK TKIP: wpapsk-tkip WPA2-PSK AES: wpa2psk-aes WPA2-PSK TKIP: wpa2psk-tkip WPA/WPA2-PSK automatic: wpawpa2psk-auto
wlan_key	Set a cipher key for wireless LAN (wlan0).

* Wireless LAN can be used by connecting a USB wireless LAN adapter compatible with USB supplied model.

Service startup setting

Item name	Description
srv_ssh	Set SSH server startup. enabled disabled
srv_ftp*	Set FTP server startup. enabled disabled
srv_samba*	Set Samba server startup. enabled disabled

*When using Ubuntu14.04 (no-SDK), the package of FTP/samba server needs to be installed.
Note that for light version rootfs, samba server is unavailable.

Rooter function setting

Item name	Description
router	Set rooter function enabled disabled
wan_if	Set WAN interface 3G: eth2 LTE: ppp0 Wireless LAN: wlan0 LAN A: eth0 LAN B: eth1
dhcp_server	Set DHCP server startup. enabled disabled
dhcp_server_lan_if	Set LAN interface of DHCP server. Wireless LAN: wlan0 LAN A: eth0 LAN B: eth1
dhcp_server_top_addr	Set DHCP initial address.
dhcp_server_alloc_num	Set the number of DHCP address allocations.

*When using Ubuntu14.04 (no-SDK), the package of DHCP server needs to be installed.

PPPoE function setting

Item name	Description
pppoe	Set PPPoE function. enabled disabled
pppoe_if	Set PPPoE interface. LAN A: eth0 LAN B: eth1
pppoe_user	Set the user name of PPPoE.
pppoe_password	Set the password of PPPoE.
pppoe_dns	Set the DNS server of PPPoE.
pppoe_firewall	Set firewall of PPPoE NONE: 0 STANDALONE: 1 MASQUERADE: 2

*PPPoE can be set with Ubuntu14.04 include SDK.
For other rootfs, PPPoE software is required separately.

Static routing function setting

Item name	Description
static_route	Set static routing function. enabled disabled
st_route_addr_1	Set the destination IP address of static routing.
st_route_gw_1	Set the gateway address of static routing.
st_route_mask_1	Set the net mask of static routing.
st_route_if_1	Set the interface of static routing.
:	
st_route_addr_32	Set the destination IP address of static routing.
st_route_gw_32	Set the gateway address of static routing.
st_route_mask_32	Set the net mask of static routing.
st_route_if_32	Set the interface of static routing.

*The number in the item names indicates setting number (up to 32).

Port forwarding function setting

Item name	Description
port_forward	Set port forwarding function. enabled disabled
port_fw_sif_1	Set port forwarding input interface.
port_fw_sport_1	Set port forwarding input port.
port_fw_daddr_1	Set port forwarding destination IP address.
port_fw_dport_1	Set port forwarding destination port.
port_fw_sif_32	Set port forwarding input interface.
port_fw_sport_32	Set port forwarding input port.
port_fw_daddr_32	Set port forwarding destination IP address.
port_fw_dport_32	Set port forwarding destination port.

* The number in the item names indicates setting number (up to 32).

IP filter function setting

Item name	Description
ipfilter	Set IP filter function. enabled disabled
ipfilter_kind_1	Set filter type. Accept Drop
ipfilter_proto_1	Set protocol. tcp, udp, icmp, all
ipfilter_saddr_1	Set the source IP address.
ipfilter_sport_1	Set the source port.
ipfilter_daddr_1	Set the destination IP address.
ipfilter_dport_1	Set the destination port.
ipfilter_kind_64	Set filter type. Accept Drop
ipfilter_proto_64	Set protocol. tcp, udp, icmp, all
ipfilter_saddr_64	Set the source IP address.
ipfilter_sport_64	Set the source port.
ipfilter_daddr_64	Set the destination IP address.
ipfilter_dport_64	Set the destination port.

*The number in the item names indicates setting number (up to 64).

For Ubuntu14.04 include SDK, the network can be set by connecting to CONPROSYS through LAN from Web browser on PC. For details, see "**Web Setup (page 57)**".

6. Driver software loading

When a driver needs manual loading or the driver that was built in "**CPS-MxS341 Series driver building (page 67)**" can be started with the command, modprobe. Run the command to the target by logging in either by serial cable connecting or through SSH.

Command example: can driver (d_can_platform) loading

```
modprobe d_can_platform
```

You can check whether the driver is operating with the next command.

Command:

```
lsmod
```

See "**Target starting sequence (page 49)**" to start the driver automatically and add the driver loading command into the "startup.sh" file.

7. Web Setup

When the rootfs type is specified as below, Ubuntu 14.04 (include SDK) contains Web setup functions.

Ubuntu 14.04 (include SDK) light (busybox)	Ubuntu 14.04 SDK within (CONPROSYS Linux SDK Self-Build edition) rootfs- light version

* For light (busybox), Web Setup function is available when Apache2 and PHP5 are set as loading tools.

This Web setup has functions such as settings of network, date & time, displays of system information, network, and more. To display the setup page, access directly the IP address of CONPROSYS from Web browser on PC.

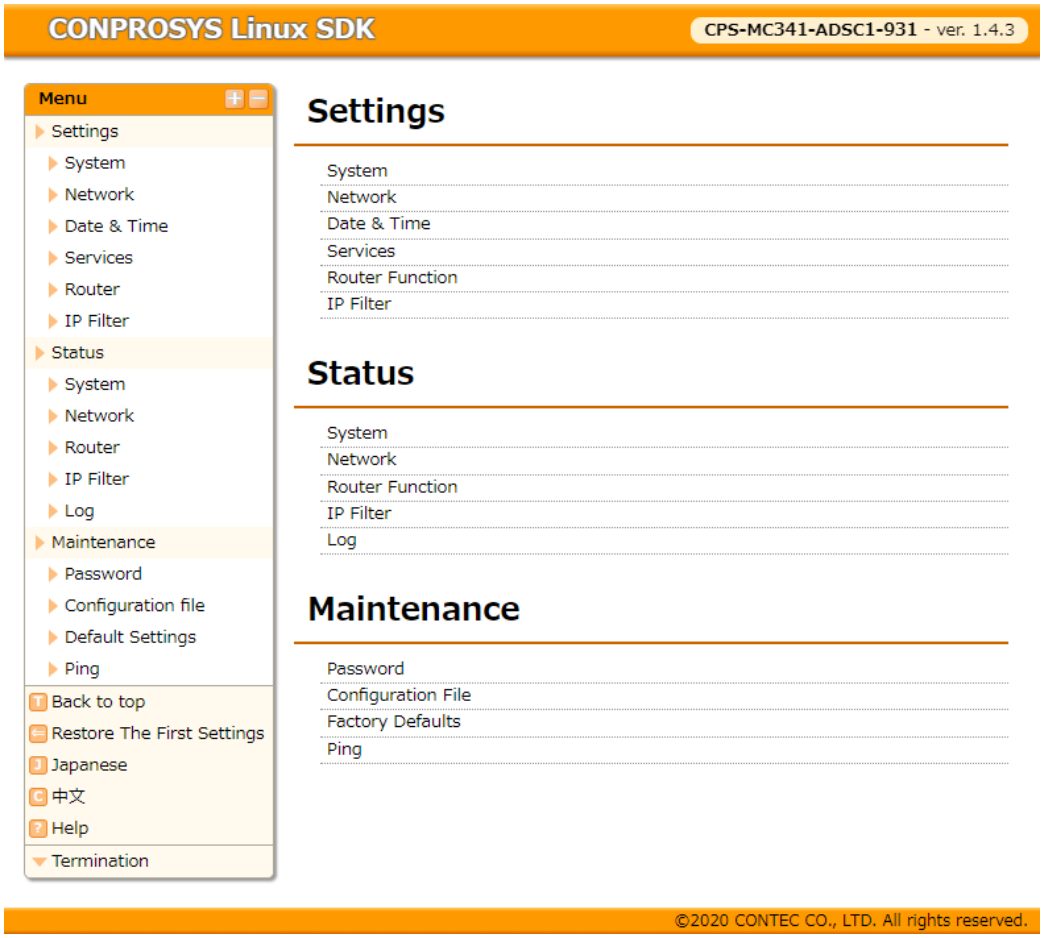
An example: Connect the PC to the LAN A port at the time of initial setting, and see whether the page can be brought in.

http://10.1.1.101/

Login: admin

Password: password

Web setup page



Web setup features following functions.

1. Setting

Each setting below can be set.

Setting menu

Setting type	Content	Default value	Note
System	Host name	(Blank)	Use the following host name when there is no item in the setting. The model name + the lower 3 bytes of MAC address
Network	Wired LAN A	10.1.1.101(fixed IP)	
	Wired LAN B	DHCP	
	3G/LTE network		3G/LTE model exclusively
	Wireless LAN	DHCP	Only when connecting with supported USB wireless adopter
Data & Time	NTP server	(Blank)	
	Manual setting		
Service	SSH server	Upon system booting: enabled	
	FTP server	Upon system booting: disabled	
	SAMBA server	Upon system booting: disabled	
Router Function	Router Function	Upon system booting: disabled	
	WAN Interface		
	DHCP Server	Upon system booting: disabled	
	Static Routing	Upon system booting: disabled	The maximum number of settings:32
	Port Forwarding	Upon system booting: disabled	The maximum number of settings:32
IP Filter	IP Filter Setting	Upon system booting: disabled	The maximum number of settings:64

2. Status

Each status below can be checked.

Status menu

Item	Description
System	This displays such as the host name, serial number, distribution/kernel information, a disk/memory usage.
Network	This displays such as IP address, MAC address, and RX/TX bytes.
Router Function	This displays is routing information.
IP Filter	This displays is IP filtering information.
Log	This displays such as syslog, other logs.

3. Maintenance

Each maintenance menu is described below.

Maintenance menu

Item	Description
Password	Password to access WEB setting page of the device can be changed.
Configuration File	The configuration file can be backed up and restored.
Default Setting	This restores the factory default setting.
Ping	Ping can be executed on the device to check network conduction.

4. Termination

Each termination menus is described below.

Termination menu

Item	Description
Save and Reboot	This saves the settings and reboots the device.
Save and Shutdown	This saves the settings and shuts down the system.
Save	This saves the settings.
Reboot	This reboots the device. If the settings are not saved before rebooting, settings return to the previous ones prior to the setting.
Shutdown	This shuts down the system. If the settings are not saved before shutting down, settings return to the previous ones prior to the setting.

For further details of each wet setup function, refer to "Help" in Web menu.

Set setting items are managed in the files below.

Configuration File:

/etc/conprosys/config.ini

Factory default setting file:

/etc/conprosys/config_def.ini

Web file is managed in the directory below.

Web content directory:

[Ubuntu 14.04 (include SDK)]

/var/www/html/

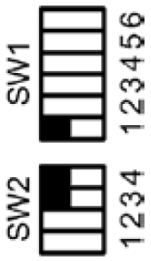
[rootfs - light version]

/opt/htdocs/

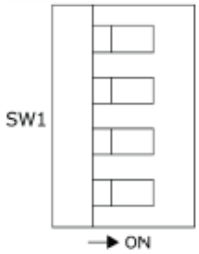
8. Initialization settings with DIP SW

With the DIP SW, restoring the factory settings of LAN A or only initializing LAN A in IP address can be implemented when the power is turned on.

Integrated type



Configurable type



DIP SW Setting Contents

SW settings	Description
Turn on SW1-2 only	Starts up IP address with the factory settings when the power is on. As for User/Password and Group settings, the system starts up with the previous settings. The present settings of IP address and User/Password can be viewed on the Web setup page.
Turn on SW1-2 and SW1-3	Restores the factory settings of individual settings when the power is on. When the restoring has finished, PWR and ST1 of LED start flashing. After confirming the flashing, turn off SW2 and SW3, then restart the system.

Building

1. Build procedure

Read each instruction to generate the run environment for the target of the CONPROSYS.

1 Build initial setting (configure.sh)
Execute ./configure.sh to generate the base of the target run environment.
Also, for the CONPROSYS Linux SDK in the self-build development, the run environment can be generated by building this initial setting.

2 The environment setting - before building (source sdkenv.txt)
Set the environment variable for building as well as generating a SD card.
After you log in, do not fail to set this before building applications.

3 Target bootloader building
Build the bootloader if you wish to change the start-up program after the power is turn on.
This usually does not need to be done.

4 Target kernel building
Build the Target kernel building to change the Linux kernel.
Perform the building if you wish to add/change/delete a driver or the kernel function without default setting. This does not need to be done if you do not change the kernel.

5 Target sample driver building
Perform the building if you wish to change the sample driver.
This does not need to be done if you do not change the sample driver.

6 Target sample library building
Perform the building if you wish to change the library.
This does not need to be done if you do not change the sample library.

7 Target sample application building
Build the sample application to run on the target of the CONPROSYS.
This does not need to be done if you do not use the sample application or develop the software in the self-build development on CONPROSYS.

2.Target bootloader building

Bootloader building can be set in a u-boot directory when you need to change the source code or the compiler option. This usually does not need to be done.
The u-boot has two types of start-up, SD card booting and built-in NOR FLASH booting.

1. Building for SD card booting

Change directory command:

```
cd ${CPS_SDK_ROOTDIR}/u-boot
```

Build command:

```
make am335x_evm
```

Copy the module (MLO, u-boot.img) generated by building with the command listed below into the boot directory for target.

Command:

```
cp -p MLO u-boot.img ${CPS_SDK_INSTALL_FULLDIR}/boot
```

If you wish to delete the object files generated by building, execute the following command.

Command:

```
make distclean
```

2. Building for built-in NOR FLASH

A bootloader module that boots from built-in NOR FLASH is different from the one booting from the SD card. Follow the instruction below to build the bootloader module.

Change directory command:

```
cd ${CPS_SDK_ROOTDIR}/u-boot
```

Build command:

```
make am335x_evm_spiboot
```

Copy the module generated by building with the command listed below into the rootfs (/InstallToMTD) for built-in NOR FLASH installation.

* To create rootfs for built-in NOR FLASH installation, please see the "**Create a rootfs section to install built-in NOR FLASH (page 38)**".

Command:

```
cp -p MLO.byteswap u-boot.img ${CPS_SDK_INSTALL_FULLDIR}/InstallerForFlash/InstallToMTD
```

3.Target kernel building

Kernel configuration/building can be set in a kernel directory when you need to change the option or implement a device driver such as unsupported USB. This usually does not need to be done.

Change directory command:

```
cd ${CPS_SDK_ROOTDIR}/kernel
```

Configuration command:

```
make menuconfig
```

Build command:

```
make uImage
```

Copy the module (uImage) generated by building with the command listed below into the boot directory for target.

Command:

```
cp -p arch/arm/boot/uImage ${CPS_SDK_INSTALL_FULLDIR}/boot/uImage.${CPS_SDK_BOARD_NAME}
```

After kernel building, build the driver module. The example command below shows copying into the rootfs directory for target.

Build command:

```
make modules
```

Example installation command:

```
sudo -E make modules_install INSTALL_MOD_PATH=${CPS_SDK_INSTALL_FULLDIR}/${CPS_SDK_ROOTFS}
```

*The driver module can be installed in the directory listed below under the rootfs directory.

lib/modules/3.2.0.CONPROSYS/

If you wish to install it into other rootfs directory, change the contents of INSTALL_MOD_PATH.

If you wish to delete the object files generated by building, execute the following command.

Command:

```
make clean
```

When the configuration information of kernel is damaged or if you wish to initialize the information, execute the following command.

Command:

```
make distclean
make ${CPS_SDK_TARGET_NAME}_defconfig
```

After executing this command, execute Kernel configuration and building.

4. CPS-MxS341 Series driver building

SDK comes with the following driver source codes of the CPS-MxS341 series in the driver directory.

- cps-driver (CPS-MxS341 System driver)
- cpsaio (CPS-MxS341 AIO driver)
- cpsdio (CPS-MxS341 DIO driver)
- 8250_cpscom (CPS-MxS341 COM driver)
- cpsssi (CPS-MxS341 SSI driver)
- cpscnt (CPS-MxS341 CNT driver)
- cps_iolib (CPS-MxS341 I/O General-purpose access driver)

Build a driver when a sample driver changed. This usually does not need to be done.

Execute kernel building beforehand since the source code from the result is necessary to build the driver. Refer to "**Target kernel building (page 65)**".

With the command below, building can be executed under each directory.

Command:

```
make
```

See below as an example of copying into the rootfs directory directory for targeting after driver building.

Command example:

```
sudo -E make modules_install INSTALL_MOD_PATH=${CPS_SDK_INSTALL_FULLDIR}/${CPS_SDK_ROOTFS}
```

* A driver module is installed into the directory listed below under target rootfs.

  lib/modules/3.2.0.CONPROSYS/extra

If you wish to install it into other rootfs directory, change the contents of INSTALL_MOD_PATH.

Note: Supplied driver software does not apply to some models.
Check the model and apply accordingly.

5. Target sample library building

SDK comes with the following source codes of shared library (shared object).

- libCpsEeprom (EEPROM data access module library)
- libCpsAio (CPS-MxS341 AIO library)
- libCpsDio (CPS-MxS341 DIO library)
- libCpsSsi (CPS-MxS341 SSI library)
- libCpsCnt (CPS-MxS341 CNT library)
- libconexio (CPS-MC341Q-ADSC1 920MHz module library)
- SerialFunc (CPS-MC341Q-ADSC1 serial module library)

Build a library when a sample library changed. This usually does not need to be done.

With the command below, library can be executed under each directory.

Command:

```
make
```

After the library building, copy into the rootfs directory for target with the following command.

Command:

```
sudo make install TARGET_ROOTFS=${CPS_SDK_INSTALL_FULLDIR}/${CPS_SDK_ROOTFS}
```

* Library module is installed into the directory listed below under target rootfs.

usr/local/lib

If you wish to install it into other rootfs directory, change the contents of TARGET_ROOTFS.

Refer to makefile or source code to create a library.

Note 1: Supplied library software does not apply to some models.
Check the model and apply accordingly.

Note 2: Each sample library software requires the supplied driver software.
See the following dependencies to apply accordingly.

libCpsAio.so ---- cpsaio.ko ---- cps-driver.ko
libCpsDio.so ---- cpsdio.ko ---- cps-driver.ko
libCpsSsi.so ---- cpsssi.ko ---- cps-driver.ko
libCpsCnt.so ---- cpscnt.ko ---- cps-driver.ko

6. Build procedure

SDK contains the sample application source codes with the following directory.

${CPS_SDK_ROOTDIR}/application/sample/

Sample applications are supplied per target. ("**Available Sample program"**)
Please apply them for a target movement evaluation or application references.
Run the "make" command per directory to generate an executable file.

An example: Timer sample program

```
cd ${CPS_SDK_ROOTDIR}/application/sample/timer
make
```

Available Sample program

Sample program	Directory application/sample/	CPS-MC341-ADSCx CPS-MC341G-ADSC1	CPS-MC341Q-ADSC1	CPS-MC341-Ax	CPS-MC341-DSx	CPS-MC341-DS1x	CPS-MCS341-DSx CPS-MGS341-DS1 CPS-MCS341G-DS1	CPS-MCS341Q-DS1
TCP/IP server/client	socket	○	○	○	○	○	○	○
Timer	timer	○	○	○	○	○	○	○
EEPROM data read	getEepromData	○	○	○	○	○	○	○
CAN transmission/reception test	can			Δ	Δ			
RS-485 communication (Integrated type)	RS485	○	○	Δ	○	○		
DI/DO, AI control (Multi-function model)	mc341_io	○	○					
AI/AO control (Integrated type)	mc341-ax_aio			○				
AI/AO control (Configurable Type)	mcs341_aio						○	○
DI/DO control (Integrated type)	spitest	○	○		○	○		
http control (DIO) (Integrated type)	http_post	○	○		○	○		
DI/DO control (Configurable Type)	mcs341_dio						○	○
SSI control (Configurable Type)	mcs341_ssi						○	○
COM control (Configurable Type)	mcs341_com						○	○
CNT control (Configurable Type)	mcs341_cnt						○	○
System control (Configurable Type)	mcs341_system						○	○
iolib control (Configurable Type)	mcs341_iolib						○	○
920MHz transmission/reception test	conexio_CMM920		○					

○: Available - Δ: Available with specific models - Blank: not available

These sample applications cannot apply to some of the CONPROSYS models.
Even with the supported products, some programs might not run due to the device port differences. Please check the program to build and test.
When building sample software that requires driver/library, build the necessary driver/library first, then build the sample application.
See Appendix "**Device I/F (page 79)**" regarding the device port.
Refer to the sample makefile or source code when you create application software.

7. Light version rootfs building

This SDK provides rootfs source code by linux light version that is utilized for small capacity booting device such as built-in NOR FLASH.　See the contents of the rootfs below.

busybox, glibc, dropbear (light version SSH server/client) , iptables, sudo

It can be built under the rootfs directory.

Change directory command:

```
cd ${CPS_SDK_ROOTDIR}/rootfs/light
```

Build command:

```
make
```

Make the rootfs file system in the rootfs under the target directory with the following command.

Command:

```
make install
```

Rootfs can be also created under the directory by giving directory that is specified as the environment variable EXPORTDIR in the Makefile.

Command example: Command example: When you wish to create the rootfs under the target directory for Built-in NOR FLASH booting.

```
make install EXPORTDIR=${CPS_SDK_ROOTDIR}/ramdisk/export
```

8. Built-in NOR FLASH booting ramdisk.xz

rootfs of built-in NOR FLASH is compressed and stored in the file named "ramdisk.xz".
With the building of light version rootfs succeeded, ramdisk.xz can be generated from its rootfs using the command below.

Change directory command:

```
cd $CPS_SDK_ROOTDIR/ramdisk
```

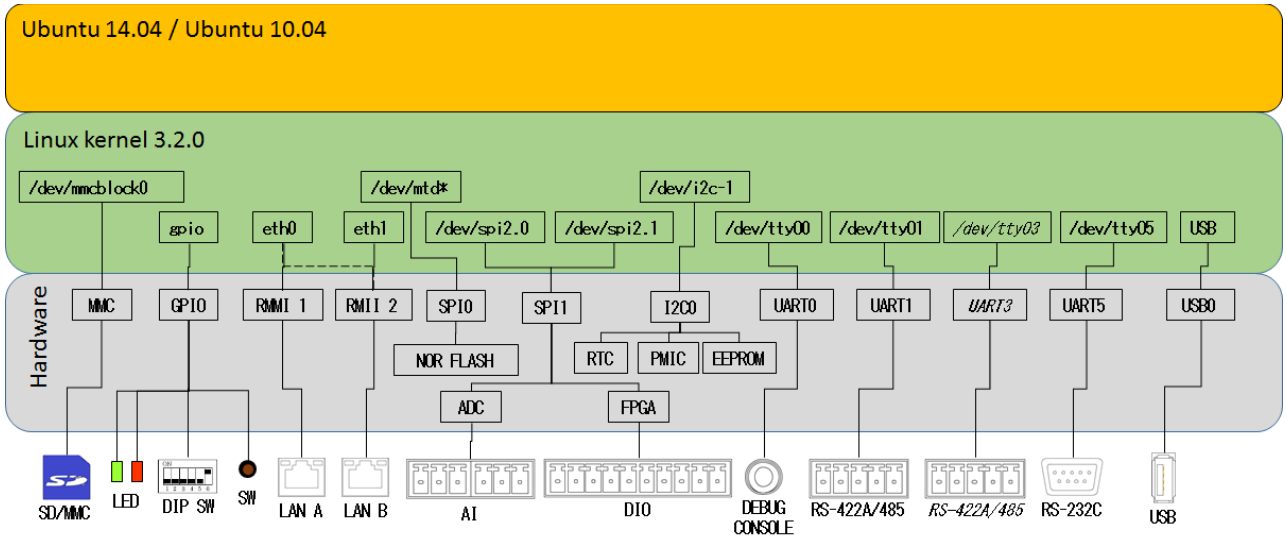
Build command:

```
make
```

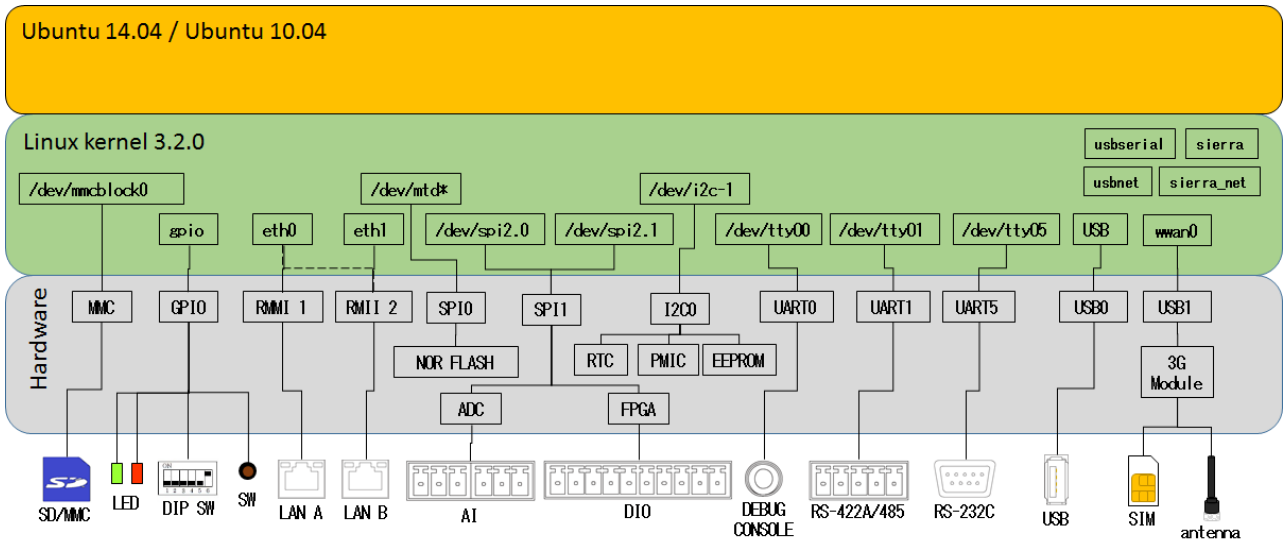
Appendix

1. Block diagram

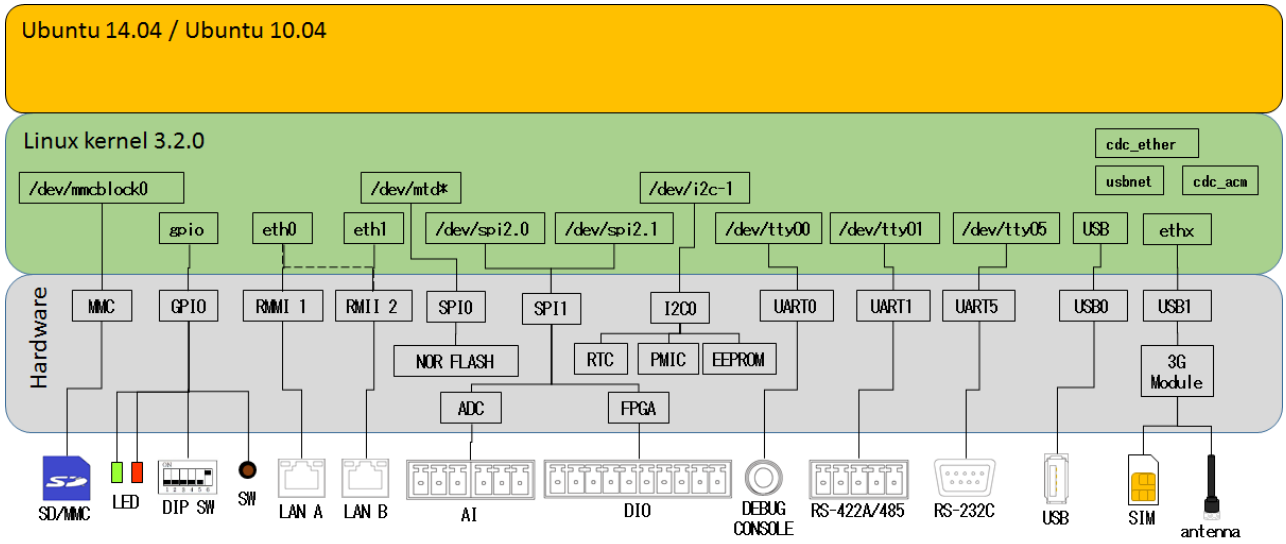
CPS-Mx341-ADSCx series block diagram *(Italic font means optional choices)*



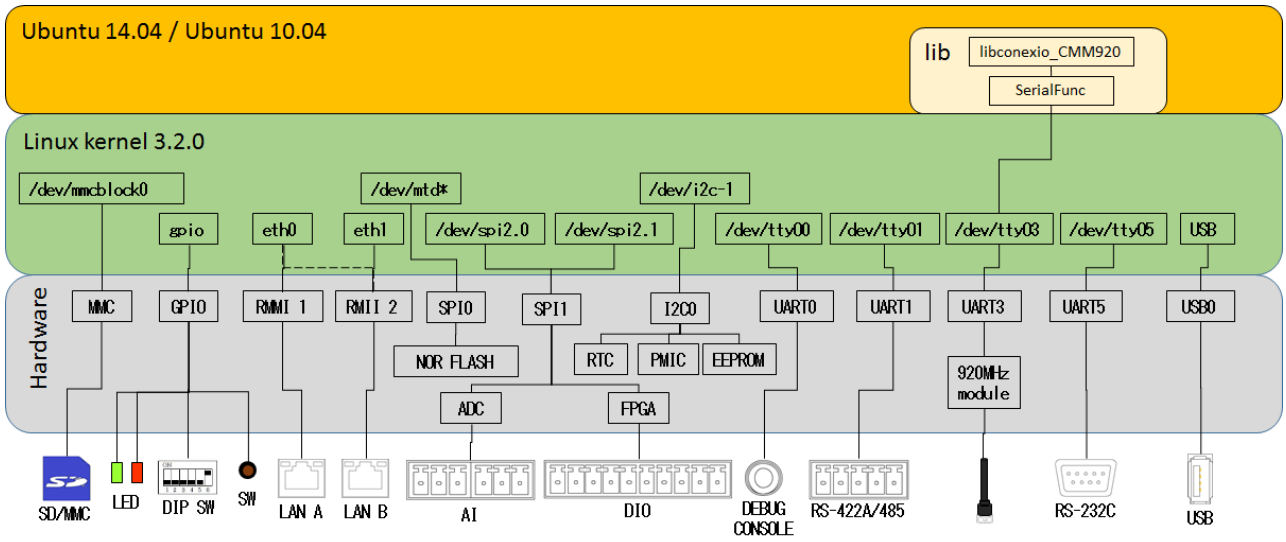
CPS-Mx341G-ADSC1 (Japan domestic model) block diagram



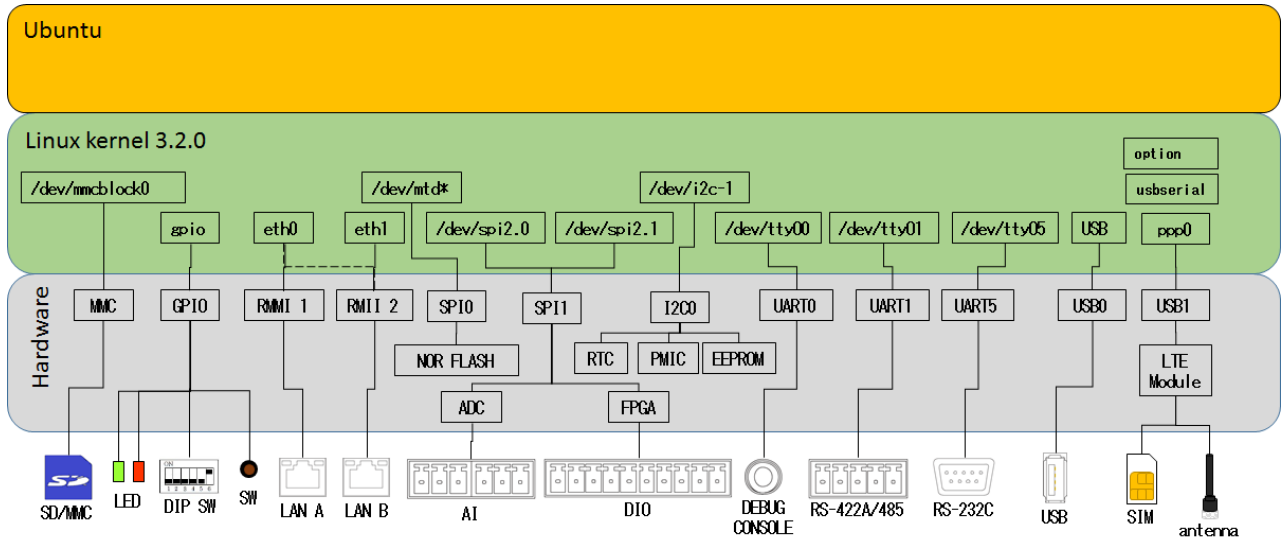
CPS-Mx341G-ADSC1 (Global model) block diagram



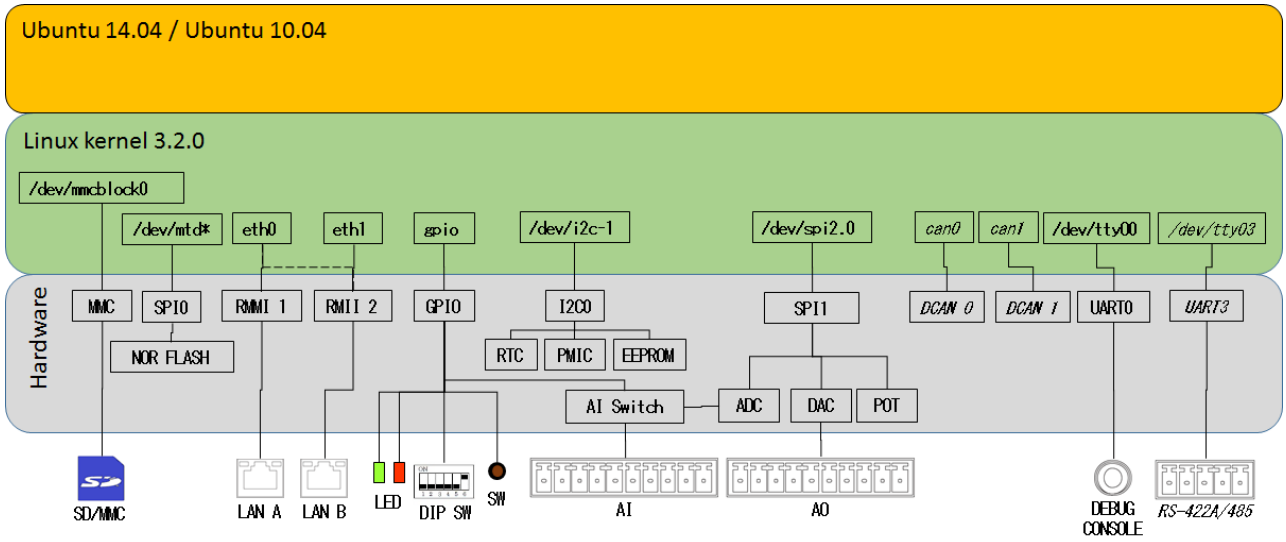
CPS-MC341Q-ADSC1 block diagram



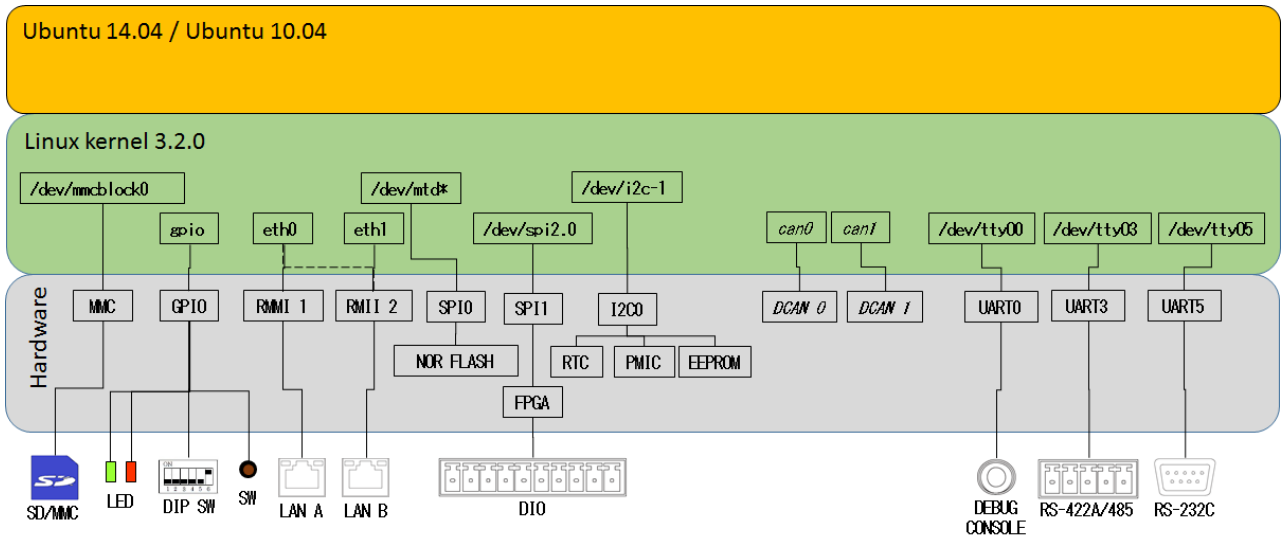
CPS-MG341G5-ADSC1 block diagram



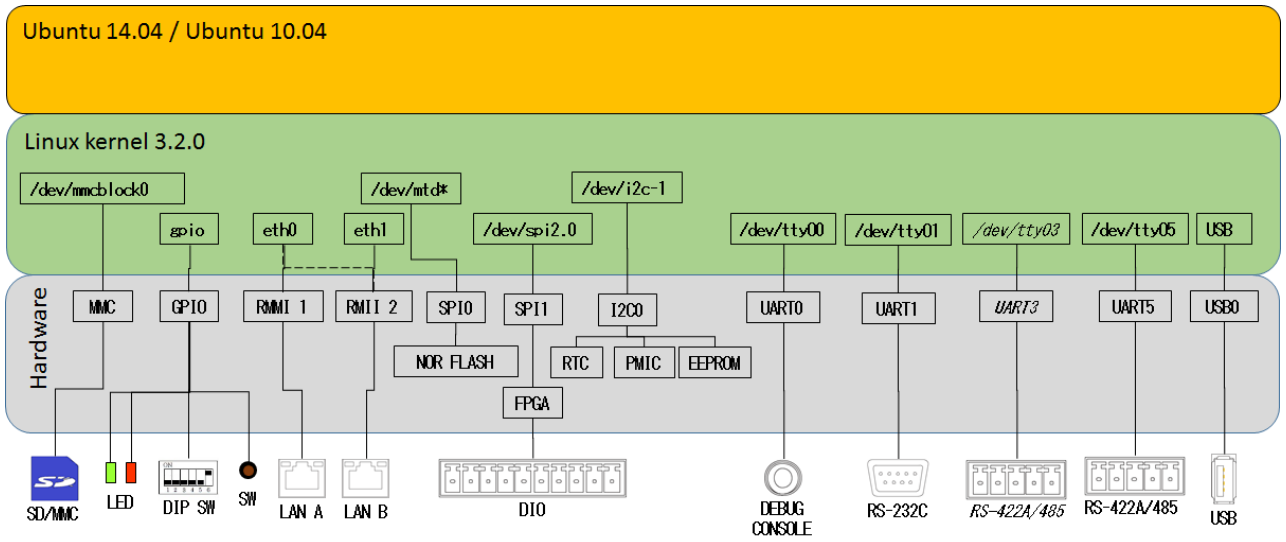
CPS-MC341-Ax series block diagram *(Italic font means optional choices)*



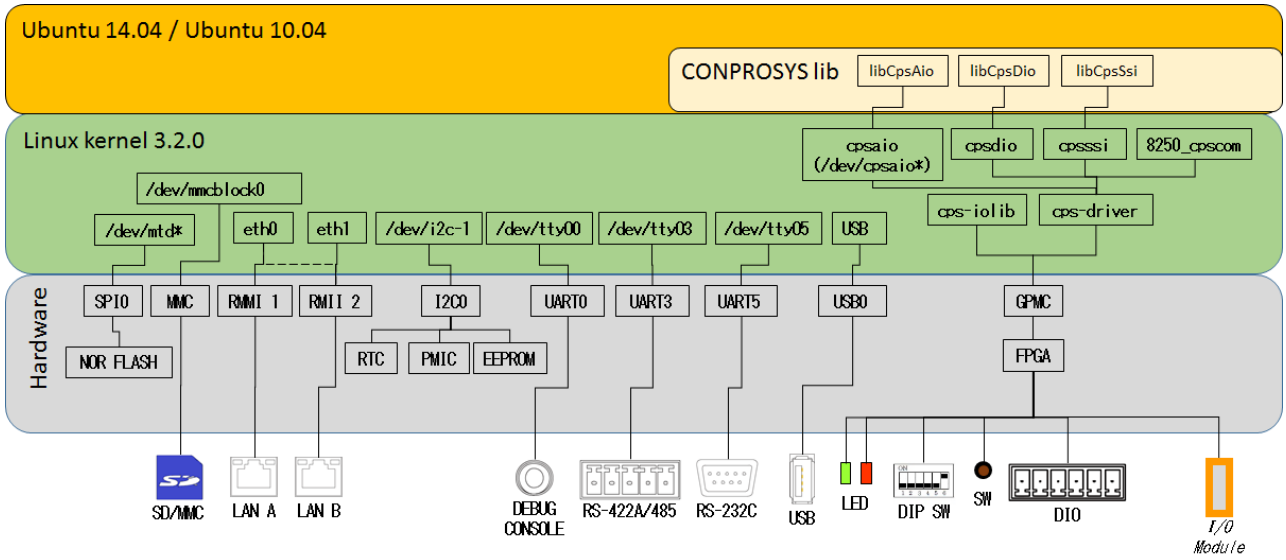
CPS-MC341-DSx series block diagram *(Italic font means optional choices)*



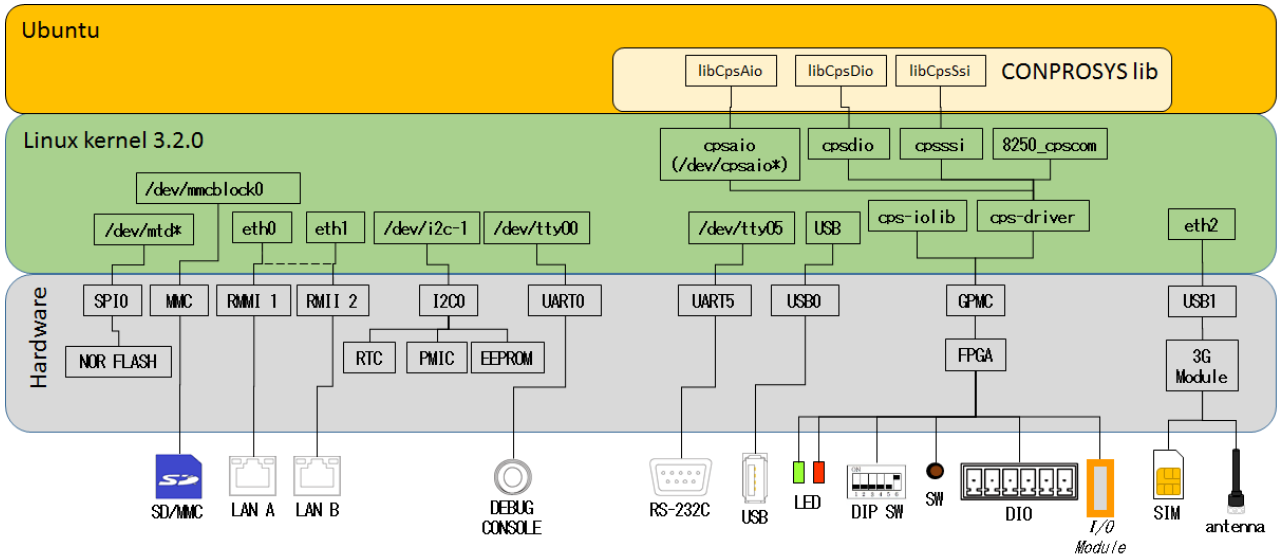
CPS-MC341-DS1x series block diagram *(Italic font means optional choices)*



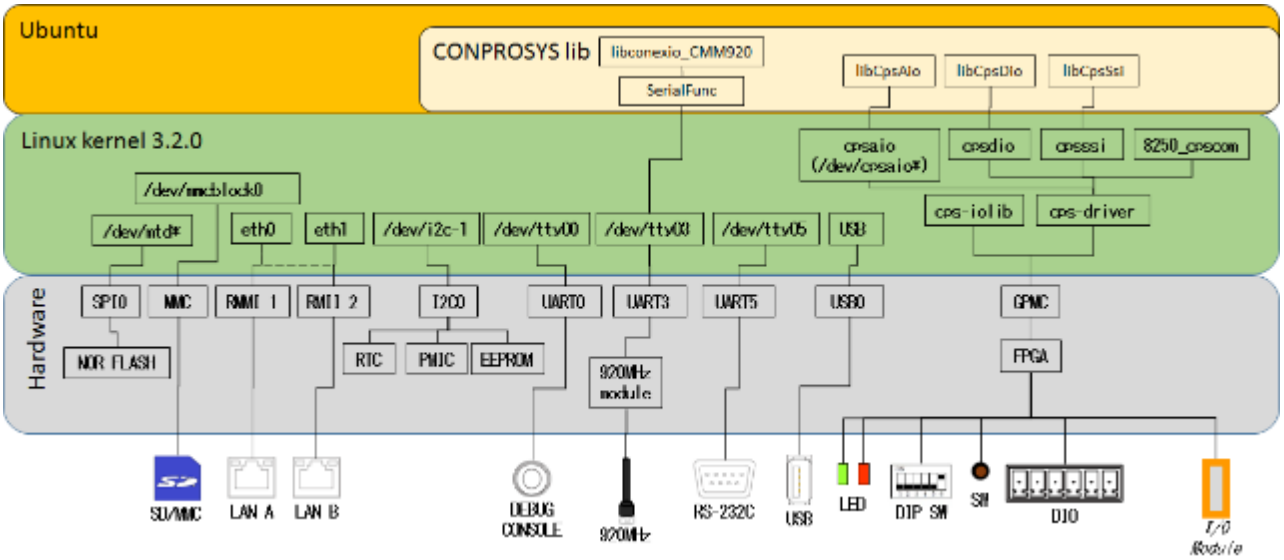
CPS-MxS341-DSx series block diagram *(Italic font means optional choices)*



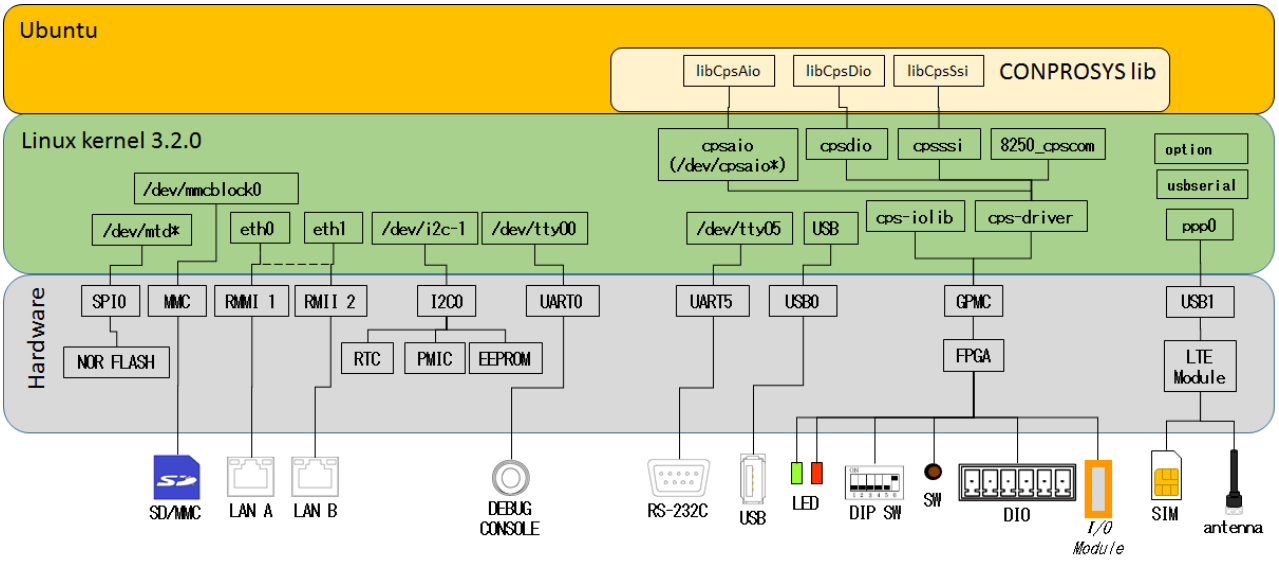
CPS-MCS341G-DS1 series block diagram *(Italic font means optional choices)*



CPS-MCS341Q-DS1 series block diagram *(Italic font means optional choices)*



CPS-MxS341G5-DS1 series block diagram *(Italic font means optional choices)*



2.Device I/F

The distinctive device I/F of CONPROSYS can be accessed on Linux as shown in the table below. Ports are vary depending on the models.

UART control device

Model	/dev/ttyO1	/dev/ttyO2	/dev/ttyO3	/dev/ttyO4	/dev/ttyO5
CPS-MC341-ADSC1	RS-422A/485 (COM A)	-	-	-	RS-232C (COM B)
CPS-MC341-ADSC2	RS-422A/485 (COM A)	-	RS-422A/485 (COM C)	-	RS-232C (COM B)
CPS-MC341G-ADSC1 CPS-MG341G5-ADSC1	RS-422A/485 (COM A)	-	-	-	RS-232C (COM B)
CPS-MC341Q-ADSC1	RS-422A/485 (COM A)	-	920MHz module	-	RS-232C (COM B)
CPS-MC341-A1	-	-	-	-	-
CPS-MC341-DS1	-	-	-	-	RS-422A/485 (COM A)
CPS-MC341-DS2	(CAN用)*1	-	-	-	RS-422A/485 (COM A)
CPS-MC341-DS11	RS-232C (COM A)	-	-	-	RS-422A/485 (COM B)
CPS-MCS341-DS1 CPS-MGS341-DS1	-	-	-	-	RS-232C
CPS-MCS341G-DS1 CPS-MCS341G5-DS1 CPS-MGS341G5-DS1	-	-	-	-	RS-232C
CPS-MCS341Q-DS1	-	-	920MHz module	-	RS-232C

*1 Reserved for CAN port.　Controlling should be operated with Network devices.

SPI control device

モデル	/dev/spidev2.0	/dev/spidev2.1	/dev/spidev2.2
CPS-MC341-ADSCx CPS-MC341G-ADSC1 CPS-MG341G5-ADSC1 CPS-MC341Q-ADSC1	AI (ADC / CLK=6MHz)	DIO (FPGA / CLK=24MHz)	-
CPS-MC341-A1	AI (ADC / CLK=6MHz)	AO (DAC / CLK=20MHz)	Potentiometers (CLK=25MHz)
CPS-MC341-DSx	DIO (FPGA / CLK=24MHz)	-	-
CPS-MC341-DS11	DIO (FPGA / CLK=24MHz)	-	-
CPS-MCS341-DS1 CPS-MGS341-DS1 CPS-MCS341G-DS1 CPS-MCS341G5-DS1 CPS-MGS341G5-DS1 CPS-MCS341Q-DS1	-	-	-

Devices and the maximum clock values of SPI controlling are listed in the parentheses.

GPIO control device (LED)

Model	GPIO 26	GPIO 27	GPIO 67	GPIO 128	GPIO 129
CPS-MC341-ADSCx CPS-MC341G-ADSC1 CPS-MC341Q-ADSC1	ST1 Green (Out)	ST2 Red (Out)	Power (Out)	-	-
CPS-MG341G5-ADSC1	ST1 Green (Out)	ST2 Red (Out)	Power (Out)	LTE Green (Out)	LTE Red (Out)
CPS-MC341-A1	ST1 Green (Out)	ST2 Red (Out)	Power (Out)	-	-
CPS-MC341-DSx	ST1 Green (Out)	ST2 Red (Out)	Power (Out)	-	-
CPS-MC341-DS11	ST1 Green (Out)	ST2 Red (Out)	Power (Out)	-	-
CPS-MCS341-DS1 CPS-MGS341-DS1 CPS-MCS341G-DS1 CPS-MCS341Q-DS1	-	-	-	-	-
CPS-MCS341G5-DS1 CPS-MGS341G5-DS1	-	-	-	LTE Green (Out)	LTE Red (Out)

GPIO control device (Switch)

Model	GPIO 32	GPIO 33	GPIO 34	GPIO 35	GPIO 87
CPS-MC341-ADSCx CPS-MC341G-ADSC1 CPS-MG341G5-ADSC1 CPS-MC341Q-ADSC1	DIP SW1-2 (In)	DIP SW1-3 (In)	DIP SW1-4 (In)	Shutdown SW (In)	-
CPS-MC341-A1	DIP SW1-2 (In)	DIP SW1-3 (In)	DIP SW1-4 (In)	Shutdown SW (In)	-
CPS-MC341-DSx	DIP SW1-2 (In)	DIP SW1-3 (In)	DIP SW1-4 (In)	Shutdown SW (In)	-
CPS-MC341-DS11	DIP SW1-2 (In)	DIP SW1-3 (In)	DIP SW1-4 (In)	Shutdown SW (In)	-
CPS-MCS341-DS1 CPS-MGS341-DS1 CPS-MCS341G-DS1 CPS-MCS341G5-DS1 CPS-MGS341G5-DS1 CPS-MCS341Q-DS1	-	-	-	-	Shutdown SW (In)

GPIO control device (Input Switch control)

Model	GPIO 39	GPIO 44	GPIO 45	GPIO 46	GPIO 47	GPIO 100
CPS-MC341-ADSCx CPS-MC341G-ADSC1 CPS-MG341G5-ADSC1 CPS-MC341Q-ADSC1	-	-	-	-	-	-
CPS-MC341-A1	DAC LDACB (Out)	AI switches A0 (Out)	AI switches A1 (Out)	AI switches A2 (Out)	AO Switch (Out)	Potentiometers $\overline{CS}$ (Out)
CPS-MC341-DSx	-	-	-	-	-	-
CPS-MC341-DS11	-	-	-	--	-	-
CPS-MCS341-DS1 CPS-MGS341-DS1 CPS-MCS341G-DS1 CPS-MCS341G5-DS1 CPS-MGS341G5-DS1 CPS-MCS341Q-DS1	-	-	-		-	-

GPIO control device (Board control)

Model	GPIO 22	GPIO 23	GPIO 36	GPIO 37	GPIO 105
CPS-MC341-ADSC1	-	-	-	-	Power RESET (Out)
CPS-MC341-ADSC2	-	-	RS485 Power (Out)	-	Power RESET (Out)
CPS-MC341G-ADSC1	-	LDO_SHUTDOWN (Out)	3G Power (Out)	3G Reset (Out)	Power RESET (Out)
CPS-MG341G5-ADSC1	PWR_ON_N_3V3 (Out)	PWRKEY (Out)	LTE Power (Out)	LTE Reset (Out)	Power RESET (Out)
CPS-MC341Q-ADSC1	-	-	920M Power (Out)	920M Reset (Out)	Power RESET (Out)
CPS-MC341-A1	-	-	-	-	Power RESET (Out)
CPS-MC341-DSx	-	-	-	-	Power RESET (Out)
CPS-MC341-DS11	-	-	-	-	Power RESET (Out)
CPS-MCS341-DS1 CPS-MGS341-DS1 CPS-MCS341G-DS1 CPS-MCS341G5-DS1 CPS-MGS341G5-DS1 CPS-MCS341Q-DS1	-	-	-	-	Power RESET (Out)

USB-Serial control device

Model	/dev/ttyUSB0	/dev/ttyUSB1	/dev/ttyUSB2	/dev/ttyUSB3	/dev/ttyUSB3
CPS-MC341-ADSCx CPS-MC341Q-ADSC1	Optional Serial Device				
CPS-MC341G-ADSC1 (Japan domestic model)	Sierra USB modem	Sierra USB modem	Sierra USB modem	Sierra USB modem	Optional Serial device
CPS-MC341G-ADSC1 (Global model)	Optional Serial device				
CPS-MG341G5-ADSC1	Quectel USB modem	Quectel USB modem	Quectel USB modem	Quectel USB modem	Optional Serial device
CPS-MC341-A1					
CPS-MC341-DSx					
CPS-MC341-DS11	Optional Serial device				
CPS-MCS341-DS1 CPS-MGS341-DS1 CPS-MCS341G-DS1 CPS-MCS341Q-DS1	Optional Serial device				
CPS-MCS341G5-DS1 CPS-MGS341G5-DS1	Quectel USB modem	Quectel USB modem	Quectel USB modem	Quectel USB modem	Optional Serial device

Integrated type ADC / DAC / FPGA (DIO) devices

Model	Device	Maker	Device model number	Control port
CPS-MC341-ADSC1 CPS-MC341-ADSC2 CPS-MC341G-ADSC1 CPS-MC341Q-ADSC1 CPS-MG341G5-ADSC1	ADC	Analog Devices	ADC7327	/dev/spidev2.0
	FPGA (DIO)	Lattice Semiconductor	LCMXO2-640HC-4TG100I	/dev/spidev2.1
CPS-MC341-A1	ADC	Texas Instruments	ADS8326IDGKR	/dev/spidev2.0
	AI Multiplexers	Analog Devices	ADG508FBRNZ	A0: GPIO 44 A1: GPIO 45 A2: GPIO 46
	DAC	Texas Instruments	DAC161S055CISQ	/dev/spidev2.1 LDACB: GPIO 39
	AO Switch	Toshiba	SSM3J135TU	Gate: GPIO 47
	Potentiometers	Analog Devices	AD5206BRUZ10	/dev/spidev2.2 $\overline{CS}$: GPIO 100
CPS-MC341-DSx	FPGA (DIO)	Lattice Semiconductor	LCMXO2-640HC-4TG100I	/dev/spidev2.0
CPS-MC341-DS11	FPGA (DIO)	Lattice Semiconductor	LCMXO2-640HC-4TG100I	/dev/spidev2.0

Regarding device control details of each AIO, please refer to each data sheet listed in the table above.
As for DIO device control (FPGA), please refer to the section "**FPGA I/O map (page 85)**" in Appendix.

Configurable type FPGA devices

Model	Device	Maker	Device model number	Control port
CPS-MCS341-DS1 CPS-MGS341-DS1 CPS-MCS341G-DS1 CPS-MCS341Q-DS1 CPS-MCS341G5-DS1 CPS-MGS341G5-DS1	FPGA	Lattice Semiconductor	LCMXO2-7000HC-4FTG256I	GPMC

As for device control (FPGA), please refer to the section "**FPGA I/O map (page 85)**" In Appendix.

Configurable type COM device

Model	/dev/ttyCPS0	/dev/ttyCPS1	/dev/ttyCPS2	/dev/ttyCPS3	…	/dev/ttyCPS62	/dev/ttyCPS63
CPS-COM-1PC	RS-232C	-	RS-232C	-	…	RS-232C	-
CPS-COM-2PC	RS-232C	RS-232C	RS-232C	RS-232C	…	RS-232C	RS-232C
CPS-COM-1PD	RS-422A/485	-	RS-422A/485	-	…	RS-422A/485	-
CPS-COM-2PD	RS-422A/485	RS-422A/485	RS-422A/485	RS-422A/485	…	RS-422A/485	RS-422A/485

Configurable type AIO control device

Model	/dev/cpsaio0	/dev/cpsaio1	…	/dev/cpsaio30	/dev/cpsaio31
CPS-AI-1608LI/ CPS-AI-1608ALI	AI	AI	…	AI	AI
CPS-AO-1604LI CPS-AO-1604ALI	AO	AO	…	AO	AO

Configurable type DIO control device

Model	/dev/cpsdio0	/dev/cpsdio1	…	/dev/cpsdio30	/dev/cpsdio31
CPS-DIO-0808L/ CPS-DIO-0808BL	DIO	DIO	…	DIO	DIO
CPS-DI-16L/ CPS-DI-16RL	DI	DI	…	DI	DI
CPS-DO-16L/ CPS-DO-16RL/ CPS-RRY-4PCC	DO	DO	…	DO	DO

Configurable type SSI control device

Model	/dev/cpsssi0	/dev/cpsssi1	…	/dev/cpsssi30	/dev/cpsssi31
CPS-SSI-4P/ CPS-SSI-4C	SSI	SSI		SSI	SSI

Configurable type FPGA control device

Model	/dev/cps-iolib
CPS-MxS341-DSx CPS-MCS341G-DS1 CPS-MCS341Q-DS1 CPS-MCS341G5-DS1 CPS-MGS341G5-DS1	GPMC

Network device

Network Category	eth0	eth1	eth2	can0	can1	wwan0	ppp0
1 LAN(Hub Mode) Type	LAN A/B	-	-	-	-	-	-
2 LAN Type	LAN A	LAN B	-	-	-	-	-
CAN on-board type 1 LAN(Hub Mode) Type	LAN A/B	-	-	CAN*	CAN*	-	-
CAN on-board type 2 LAN(Hub Mode) Type	LAN A	LAN B	-	CAN*	CAN*	-	-
3G on-board type (Japan domestic model) 1 LAN(Hub Mode) Type	LAN A/B	-	-	-	-	3G	-
3G on-board type (Japan domestic model) 2 LAN Type	LAN A	LAN B	-	-	-	3G	-
3G on-board type (Global model) 1 LAN(Hub Mode) Type	LAN A/B	3G	-	-	-	-	-
3G on-board type (Global model) 2 LAN Type	LAN A	LAN B	3G	-	-	-	-
LTE on-board type 1 LAN Type	LAN A/B	-	-	-	-	-	LTE
LTE on-board type 2 LAN Type	LAN A	LAN B	-	-	-	-	LTE

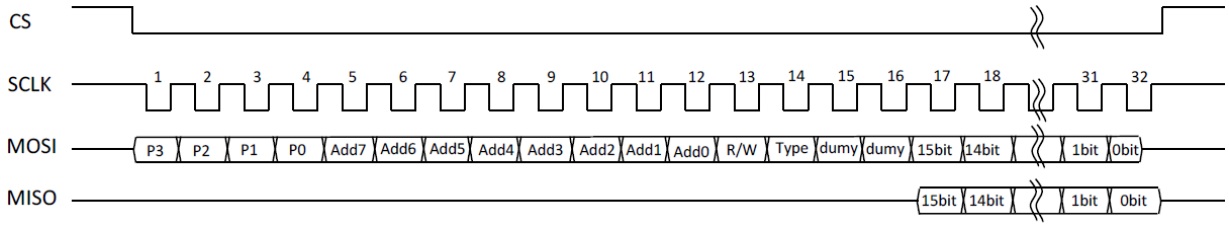
* As for CAN on-board model, load the driver (d_can_platform) for CAN to control.

3. FPGA I/O map

1. Integrated type CPS-Mx341-ADSCx / DSx Series

Maker: Lattice Semiconductor
Device model number: LCMXO2-640HC-4TG100I
Interface: SPI

SPI signal timing



MOSI: Slave latches the signal with a fall of SCLK

MISO: Output slave signals with a rise of SCLK. Master latches the signal with a fall of SCLK.

SPI signal format

Register Page	Address	R/W	Access Type	Dummy	Data
4bit	8bit	1bit	1bit	2bit	16bit

- R/W: 0 = Read, 1 = Write
- Access Type: 0 = Byte Access, 1 = Word Access
- Dummy: Always 0

Upon Byte accessing, data are LSB justified and converted into 16 bit for transmission and reception.

Example: Write 00AAh to Page = 0h, Address=12h.
0x0 12 C 00AA

Products Category

Products Category	Function	Register Page	Product
01h	Digital I/O unit	0h	CPS-MC341-ADSCx, CPS-MC341-DSx
02h	Analog input unit	1h	CPS-MC341-ADSCx
03h	Counter unit	2h	CPS-MC341-ADSCx

Digital I/O unit port map (Page 0h)

Address	Read/Write	Meaning
00h – 01h	R	System reservation area
02h – 03h	R	System reservation area
04h – 0Ch	R	Not used
0Eh – 0Fh	R	System reservation area
10h – 11h	R	Digital input port
12h – 13h	R/W	Digital output port
14h – 17h	R	Not used
18h – 19h	R/W	Digital filter setting time
1Ah – 1Fh	R	Not used
1Ch – 1Dh	R/W	Internal power ON/OFF*
1Eh – 1Fh	R	Not used
20h – 21h	R/W	System reservation area
22h – 23h	R	Not used
24h – 25h	R/W	System reservation area
26h - FFh	R	Not used

*For CPS-MC341-ADSC1-931 exclusively

Analog input unit port map (Page 1h)

Address	Read/Write	Meaning
00h – 01h	R	System reservation area
02h – 03h	R	System reservation area
04h – 27h	R	Not used
28h – 29h	R/W	Analog input unit
2Ah - FFh	R	Not used

Counter I/O unit port map (Page 2h

Address	Read/Write	Meaning
00h – 01h	R	System reservation area
02h – 03h	R	System reservation area
04h – 0Fh	R	Not used
10h – 11h	R/W	Direct Counter Data lower (R) / Read Channel Select (W)
12h – 13h	R/W	Direct Counter Data higher (R) / Direct Counter Latch Select (W)
14h – 15h	R/W	Counter Select Enable Status
16h – 17h	R	Not used
18h – 19h	R/W	Command Select
1Ah – 1Bh	R	Not used
1Ch – 1Dh	R/W	Counter Input / Output data lower data
1Eh – 1Fh	R/W	Counter Input / Output data higher data
20h – 21h	W	System reservation area
22h – 23h	W	System reservation area
24h – 25h	R/W	System reservation area
26h – 27h	R/W	System reservation area
2Ah - FFh	R	Not used

Digital input port (Page 0h / Address 10h - 11h)　R

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0

This port gets a value of digital input terminal.

When digital filter is set, a value after passing through the filter is taken.

*This function is only available with DI0 – DI3 in CPS-MC341-ADSCx series.

Digital output port (Page 0h / Address 12h -13h)　R/W

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D05	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	DO7	DO6	DO5	DO4	DO3	DO2	DO1	DO0

This port sets a value of digital output terminal or gets a setting value.

*This function is only available with DO0 – DO1 in CPS-MC341-ADSCx series.

Digital filter setting time (Page 0h / Address 18h – 19h)　R/W

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	ST4	ST3	ST2	ST1	ST0	0	0	0	0	0	0	0	0

This port sets a digital filter value applicable to digital input terminal or gets a setting value.

The setting value applies to all input terminals.　See "**Digital filter settings(page 88)**" for the setting value.

Digital filter settings

Set	Name	Meaning	Set item	Initial value
ST4 - 0	Digital filter setting time	Set the digital filter time	0: filter function not used	0 [filter function not used]
			1: 0.25µsec	
			2: 0.5µsec	
			3: 1µsec	
			4: 2µsec	
			5: 4µsec	
			6: 8µsec	
			7: 16µsec	
			8: 32µsec	
			9: 64µsec	
			10: 128µsec	
			11: 256µsec	
			12: 512µsec	
			13: 1.024msec	
			14: 2.048msec	
			15: 4.096msec	
			16: 8.192msec	
			17: 16.384msec	
			18: 32.768msec	
			19: 65.536msec	
			20: 131.072msec	
			21～31: Reserve	

Internal power ON/OFF setting port (Page 0h / Address 1Ch – 1Dh) R/W

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	PWEn

This ports sets enabled(ON) or disabled(OFF) of internal power for digital input port.

Read this port to check the setting status. See "**TInternal power ON/OFF setting (page 88)**" for setting value.

Internal power ON/OFF setting

Set	Name	Meaning	Set item	Initial value
PWEn	Internal power enabled	Set the internal power enabled (ON).	0: disabled (OFF) 1: enabled (ON)	0 [disabled]

Analog input port (Page 1h / Address 28h - 29h) R

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	03	D2	D1	00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	AT1	AT0

This port obtains a value of analog input channel. When an isolation between channels is needed, do not turn on both switches simultaneously. It disables an isolation function.

Counter data read port (Page 2h / Address 10h - 13h) R

Addr	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
10h	D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
12h	0	0	0	0	0	0	0	0	D23	D22	D21	D20	D19	D18	D17	D16

This port reads latched counter data.

Set "**Counter read channel setting port (Page 2h / Address 10h) W (page 89)**" to read data.

Counter read channel setting port (Page 2h / Address 10h) W

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Sel0

This port selects a channel to read from Counter data read port.

See the instruction "**Counter data read port (Page 2h / Address 10h - 13h) R (page 89)**" to read counter data.

Counter read setting

Set	Name	Meaning	Set item	Initial value
Sel0	Counter read channel	Set the cannel to read from counter data read port	0: Channel 0 1: Channel 1	0 [Channel 0]

Counter data latch setting port (Page 2h / Address 12h) W

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ch01	Ch00

Write "1" into this port to latch counter data. The latched count values in this section are read from the Counter data read port.

Counter valid channel setting port (Page 2h / Address 14h) R/W

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ch01	Ch00

This port sets counter valid channels and reads the status of the setting.

Counter command port (Page 2h / Address 18h) W

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	Cmd06 - 00						

This port is an operation command port to execute the following command codes.

Command code list:

08h: Ch0 counter mode (Write)

09h: Ch1 counter mode (Write)

18h: Ch0 comparison register 0 (Write)

19h: Ch1 comparison register 0 (Write)

20h: Ch0 comparison register 1 (Write)

21h: Ch1 comparison register 1 (Write)

38h: Count match status check / clear (Read/Write)

3Ah: Carry status check / clear (Read/Write)

3Dh: Zero clear (Write)

When executing the command to write, the data are set into data address port (Page 2h / Address 1Ch – 1Fh). When executing the command to read, the data are read from data address port (Page 2h / Address 1Ch – 1Fh).

Control data address port after controlling the command port.

Refer to "**Configurable type DIO control device (page 83)**" to "**Products Category (page 85)**" regarding the format of the data address port for each command code.

Ch0 / Ch1 counter mode (counter command code: 08h / 09h) W

Addr	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1Eh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This sets operation modes of the counter. Each mode can be set per input channel.

Ch0 / Ch1 comparison register 0 (counter command code: 18h / 19h) W

Addr	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1Ch	Data00 – 15															
1Eh	0	0	0	0	0	0	0	0	Data16 - 25							

This sets data into Ch0 – Ch1 count value comparison register 0.

Ch0 / Ch1 comparison register 1 (counter command code: 20h / 21h) W

Addr	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1Ch	Data00 – 15															
1Eh	0	0	0	0	0	0	0	0	Data16 - 25							

This sets data into Ch0 – Ch1 count value comparison register 1.

Count match status check / clear (counter command code: 38h) R/W

Addr	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1Ch	0	0	0	0	0	0	Cmp1_Ch1	Cmp1_Ch0	0	0	0	0	0	0	Cmp0_Ch1	Cmp0_Ch0
1Eh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Upon reading, the bit becomes 1 if the condition is satisfied.

Upon writing, set 1 into the corresponding bit to reset.

Carry status check / clear (counter command code: 3Ah) R/W

Addr	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Carry Ch1	Carry Ch0

Upon reading, the bit becomes 1 if the condition is satisfied.

Upon writing, set 1 into the corresponding bit to reset.

Zero clear (3Dh) W

Addr	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Ch1	Ch0

Upon writing, set 1 into the corresponding bit to reset.

2. Configurable type CPS-MxS341-DSx Series

Maker: Lattice Semiconductor
Device model number: LCMXO2-7000HC-4FTG256I
Interface: GPMC

Port map

Address	Read/Write	Meaning
0000h – 0001h	R	System reservation area
0002h	R	Rotary switch
0003h	R	DIP SW
0004h	R	The number of device connections
0005h	R/W	System reservation area
0006h – 0007h	R/W	LED control
0008h – 000Bh	R/W	System reservation area
000Ch – 000Dh	R/W	System reservation area
000Eh – 000Fh	R	Not used
0010h – 005Fh	R	Information of configurable type product
0060h – 0063h	R/W	DIO control register
0064h – 0065h	R/W	UART control register
0066h – 00FFh	R/W	Not used
0100h – 01FFh	R	Device 0
0200h – 02FFh	R	Device 1
: :		: :
1F00h – 1FFFh	R	Device 30
2000h – 20FFh	R	Device 31

4.Built-in NOR FLASH memory map

CONPROSYS contains 32Mbyte built-in NOR FLASH memory.

Relations between the memory layout and the installation file listed in "**Copy the results into the rootfs section for installation (page 40)**" are shown below.

NOR FLASH memory map

Address	dev	Memory map	Purpose	Installing file
0000000h – 001FFFFh	mtd0	131,072 byte	for Master booting	MLO.byteswap
0020000h – 009FFFFh	mtd1	524,288 byte	for u-boot	u-boot.img
00A0000h – 00DFFFFh	mtd2	262,144 byte	for u-boot Option	*1
00E0000h – 043FFFFh	mtd3	3,538,944 byte	for kernel	uImage
0440000h - 0DBFFFFh	mtd4	9,961,472 byte	for ramdisk	ramdisk.xz
0DC0000h – 1FFFFFFF	mtd5	19,136,512 byte	application area	mtd5.tgz*2

*1. The installation tool of this SDK default comments out processing.

*2. The file should be extracted to install into mtd5.

5. Integrated type products LED / DIP Switch / Switch control

LEDs of Integrated types listed below can be controlled by GPIO port.

LED control for integrated type

LED	Control device	Port No	Port attribute	Control method (linux shell)
Power	GPIO	67	Out	On : /usr/local/bin/gpio_out.sh 67 0 Off : /usr/local/bin/gpio_out.sh 67 1
ST1	GPIO	26	Out	On : /usr/local/bin/gpio_out.sh 26 0 Off : /usr/local/bin/gpio_out.sh 26 1
ST2	GPIO	27	Out	On : /usr/local/bin/gpio_out.sh 27 0 Off : /usr/local/bin/gpio_out.sh 27 1

Switches of integrated types listed below can be read by GPIO port.

Switch control for integrated type

LED	Control device	Port No	Port attribute	Control method (linux shell)
DIP SW1-2	GPIO	32	In	/usr/local/bin/gpio_in.sh 32 On=0, Off=1
DIP SW1-3	GPIO	33	In	/usr/local/bin/gpio_in.sh 33 On=0, Off=1
DIP SW1-4	GPIO	34	In	/usr/local/bin/gpio_in.sh 34 On=0, Off=1
Shutdown SW	GPIO	35	In	/usr/local/bin/gpio_in.sh 35 Press(On)=0, Release(Off)=1

6. Configurable type products DIO / LED / DIP Switch / Switch control

DIO / LED / DIP Switch / Switch of configurable types can be controlled by file under the directly on CONPROSYS listed below.

/sys/bus/platform/drivers/cps-driver

See the function and usage of each file in "**Configurable type DIO / LED / DIP Switch / Switch control (page 96)**".

Configurable type DIO / LED / DIP Switch / Switch control

File	Control device	Function
	How to use	
dio0_direction	DIO	DI/DO switch setting
	Set DI when b0(DIO0) – b3(DIO3) are 0, set DO when b0(DIO0) – b3(DIO3) are 1. Setting example: Set DIO0 and DIO1 to DI, DIO2 and DIO3 to DO. b3:1, b2:1, b1:0, b0:0 → cH <Command> echo 0xc > /sys/bus/platform/drivers/cps-driver/dio0_direction Setting reading example: <Command> cat /sys/bus/platform/drivers/cps-driver/dio0_direction	
dio0_do_value	DO	DO value setting
	Setting example: Set DO0 and DO2 to 1, DO1 and DO3 to 0. b3:0, b2:1, b1:0, b0:1 → 5H <Command> echo 0x5 > /sys/bus/platform/drivers/cps-driver/dio0_do_value Setting reading example: <Command> cat /sys/bus/platform/drivers/cps-driver/dio0_do_value	
dio0_di_value	DI	DI value reading
	<Command> cat /sys/bus/platform/drivers/cps-driver/dio0_di_value	
id	Rotary switch	Rotary switch value reading
	<Command> cat /sys/bus/platform/drivers/cps-driver/id	
led_status1	Status1 LED	Status1 LED On/Off setting
	Setting example: Turn on the light of Status 1LED. <Command> echo 1 > /sys/bus/platform/drivers/cps-driver/led_status1 Setting reading example: <Command> cat /sys/bus/platform/drivers/cps-driver/led_status1	
led_status2	Status2 LED	Status2 LED On/Off setting
	Setting example: Turn off the light of Status 2 LED. <Command> echo 0 > /sys/bus/platform/drivers/cps-driver/led_status2 Setting reading example: <Command> cat /sys/bus/platform/drivers/cps-driver/led_status2	
led_error	Error LED	Error LED On/Off setting
	Setting example: Turn on the light of Error LED. <Command> echo 1 > /sys/bus/platform/drivers/cps-driver/led_error Setting reading example: <Command> cat /sys/bus/platform/drivers/cps-driver/ switch	
switch	DIP Switch	DIP Switch value reading
	<Command> cat /sys/bus/platform/drivers/cps-driver/switch	

You can access the LED/Switch of the configurable type using iolib control sample program for the CPS-MxS341-DSx series included in the sample applications.

Through GPMC port, LEDs of configurable type can be controlled by FPGA I/O map addresses listed below.

LED control for configurable type

Register	D7	D6	D5	D4	D3	D2	D1	D0
0006h	-	-	-	-	ERR	ST2	ST1	Power
					R/W	R/W	R/W	R/W
					On: 1 Off: 0	On: 1 Off: 0	On: 1 Off: 0	On: 1 Off: 0

Command example: Turn on the lights of Power, ST1, and ST2.

```
gpmc_testd –w1 0006 06
```

Command example: Obtain the status of LED.

```
gpmc_testd –r1 0006
```

Through GPMC port, the switches of configurable type can be read by FPGA I/O map addresses listed below.

Switch control for configurable type

Register	D7	D6	D5	D4	D3	D2	D1	D0
0002h	Rotary switch H				Rotary switch L			
0003h	DIP SW1-4	DIP SW1-3	DIP SW1-2	DIP SW1-1	-	-	-	-
	On: 1 Off: 0	On: 1 Off: 0	On: 1 Off: 0	On: 1 Off: 0	-	-	-	-

Command example: Obtain the status of rotary switch

```
gpmc_testd –r1 0002
```

7. Option Board control

The models listed below contain an option board of 3G/LTE/920Hz communication.

[Integrated type products M2M Controller Series]

CPS-MC341G-ADSC1 Series　Multi-I/O + 3G WAN (Japan domestic / Global) Model
CPS-MC341Q-ADSC1　Multi-I/O + 920MHz (Japan only) Model

[Integrated type products M2M Gateway Series]

CPS-MG341G-ADSC1 Series　Multi-I/O + 3G WAN (Japan only) Model
CPS-MG341G5-ADSC1　Multi-I/O + LTE Model

[Configurable type products M2M Controller Series]

CPS-MCS341G-DS1　Controller + 3G WAN (Japan only) Model
CPS-MCS341G5-DS1　Controller + LTE Model
CPS-MCS341Q-DS1　Controller + 920MHz (Japan only) Model

[Configurable type products M2M Gateway Series]

CPS-MGS341G5-DS1　Controller + LTE Model

These models can control the power of the option board.

Option board control

Function	How to control (linux shell)
Option board power On*	/usr/local/cps-board/PowerOnOptionBoard.sh
Option board power Off*	/usr/local/cps-board/PowerOffOptionBoard.sh
Option board detection	/usr/local/cps-board/DetectOptionBoard.sh [End Status] 0: Option board activated 1: Option board no detection

*root privileges is requested. Use the sudo command when controlling the power in the console.

The models with 3G/LTE can control such as connection/disconnection, SIM check, and RSSI acquisition.

3G/LTE control

Function	How to control (linux shell)
Connection *1	/usr/local/cps-board/mobile/start_mobile.sh
Disconnection *1	/usr/local/cps-board/mobile/stop_mobile.sh
3G/LTE module reset *1	/usr/local/cps-board/mobile/reset_mobile.sh
SIM check	/usr/local/cps-board/mobile/checkSIM_mobile.sh [End status] 0: When SIM is detected, this displays the "Detect SIM" 1: When SIM is not detected, this displays the "Not Detect"
RSSI acquisition	/usr/local/cps-board/mobile/checkSIM_mobile.sh [End status] 0: Succeed displays RSSI value (dbm) 1: Fail
RSRP acquisition (only for the models with LTE)	/usr/local/cps-board/mobile/getRSRP.sh [End status] 0: Succeed displays RSRP value (dbm) 1: Fail
Option board LED control *2	/usr/local/cps-board/mobile/ctrl_LED.sh param [param] 0: All off 1: Green On Red Off Red Off 2: Green Off Red On Red On 3: Green On Red On Red On [End status] 0: Succeed 1: Fail

*1. *root privileges is requested. Use the sudo command when controlling the power in the console.

*2. As for the CPS-MC341G-ADSC1-111 and CPS-MG341G-ADSC1-111 models, LED control is not available since 3G module control is used in these products.

8. Target on-board application

Primary on-board applications

Application	Light rootfs NOR Flash version	Light rootfs SD version	Ubuntu 14.04	Ubuntu 14.04 with SDK
busybox	1.31.1	1.31.1	-	-
apt-utils	-		1.0.1	1.0.1
binutils	-		-	2.24-5
ncurses	-		-	5.9
apache	2.4.29 *1	2.4.29 *1	-	2.4.7-1
ssh server/client	dropbear 2019.78	dropbear 2019.78	open-ssh 6.6	open-ssh 6.6
NTP client	(busybox)	(busybox)	ntpdate 4.2.6	ntpdate 4.2.6
DHCP server	(busybox)	(busybox)	Udhcpd 1.21.0-1	isc-dhcp-server 4.2.4-7
DHCP client	(busybox)	(busybox)	isc-dhcp-client 4.2.4-7	isc-dhcp-client 4.2.4-7
Samba server	-	-	-	4.3.11
Samba client	-	-	-	4.3.11
Nfs Server	-	-	-	-
Nfs Client	-	-	-	-
gcc / g++	-	-	-	4.9.4-2
cmake	-	-	-	3.2.2-2
autoconf	-	-		2.69-6
automake	-	-		1.14.1-2
perl	-	-	5.18.2	5.18.2-2
python	-	-	3.4.3-1	3.4.3-1
php5	5.6.34 *1	5.6.34 *1	-	5.5.9
curl	7.59.0 *1	7.59.0 *1	7.35.0-1	7.35.0-1
wget	(busybox)	(busybox)	1.15	1.15-1
ftp server	(busybox)	(busybox)	-	vsftpd 3.0.2
ftp client	(busybox)	(busybox)	-	0.17
tftp server	(busybox)	(busybox)	-	-
tftp client	(busybox)	(busybox)	-	-
mail	(busybox)	(busybox)	-	-
iperf	-	-	-	-
minicom	-	-	-	-
ppp	2.4.7	2.4.7	2.4.5-5.1	2.4.5-5.1
pppoe	-	-	-	3.8-3
iptables	1.8.4	1.8.4	1.4.21-1	1.4.21-1
Wireless tool	29 *1	29 *1	30~pre9-8	30~pre9-8
wpa_supplicant	2.7 *1	2.7 *1	2.1-0	2.1-0
Open SSL	1.0.2n *1	1.0.2n *1	1.0.1f-1	1.0.1f-1
sudo	1.8.31p1	1.8.31p1	1.8.9p5-1	1.8.9p5-1
gdb	-	-	-	8.2

*1 Optional

Revision History

MONTH YEAR	Summary of Changes
April 2016	1st edition
April 2016	- Modified the directory name mistake of the SD of rootfs for the installation to FLASH ROM ([4-2. Write installing software on a SD card for built-in NOR FLASH]) - Append the recommended the serial monitor cable model number to "[Required items for development]" - Append [Target sample library building] - Append the sample application program below for Configurable type AI/AO control, DI/DO control, SSI control
October 2016	Ver 1.1.0 - Added new models CPS-MC341G-ADSC1 series (global model) CPS-MC341Q-ADSCx series, CPS-MC341-DS1x series - Clarified the availability of SD card (available with SDHC unavailable for SDXC) - Discontinued the 32bit version Ubuntu14.04 in a required specification of the development host PC - Changed SDK installation means - Added the flow chart of the building procedure - Supported CONPROSYS Ubuntu14.04 - Added "Create a SD image file" in the Create a SD card for start-up - Appendix, modified certain contents in device I/F [GPIO control device] - Appendix, added USB-Serial control device in the device I/F - Appendix, modified contents of the target on-board application
October 2017	Ver 1.2.0 - Added new models CPS-MCS341G-DSx series CPS-MCS341Q-DSx series - Added I/O support for the following modules CPS-AI-1608ALI, CPS-CNT-3202I - Added the internal power ON/OFF setting board for CPS-MC341-ADSC1-931 - Added Web setup function (Ubuntu 14.04 include SDK version exclusively) - Changed how to set the network setting - Added Ubuntu16.04 (64bit version) in a required specification of the development host PC - Combined the description of 3G model settings into "5-5. Target network setting" - Combined the contents of appendix "E.LED control" and "F.DIP Switch/Switch", and reorganized them with the titles as "E. Integrated type products LED / DIP Switch / Switch control" and "F. Configurable type products DIO / LED / DIP Switch / Switch control" - Added " Create a SD card for built-in NOR FLASH installation (Create a SD image file)" - Stopped using –m and -t options in configure.sh
May 2018	Ver 1.3.0 - Added sudo, iptables in light version rootfs

MONTH YEAR	Summary of Changes
	- Added the item choice to set Wireless tool, Apache in light version rootfs. When Apache is set, Web Setup function is also included - Added rooter function and IP filter function in Network setting item and Web Setup functions - Upgraded " Primary on-board applications" - Added I/O support for the following modules CPS-AI-1608ALI, CPS-CNT-3202I - Added a driver and a library for the configurable controller cpscnt (CNT driver for CPS-MCS341 libCpsCnt (CNT library for CPS-MCS341)
August 2020	Ver 1.4.0 - Added new models CPS-MG341-ADSC1 series, CPS-MG341G-ADSC1 series CPS-MG341G5-ADSC1, CPS-MCS341G5-DS1 - Updated the model names of the [List of CONPROSYS products that support SDK] - Divided the Build section into the [Cross-Build Environment] and the [Building] - Changed the figure in [Target starting sequence] - Changed building in the [Light version rootfs building] - Updated the list of the tables in the [Appendix - Device I/F] - Added the new section of the [Appendix - Option Board control] - Updated the list of [8. Target on-board application] - Consolidated the CONPROSYS products into five types in the section [Cross-Build Environment - 2. Initial Settings] - Excluded Ubuntu 10.04 from the target rootfs type
February 2022	Ver 1.4.3 - Added the new section of the [Target operation check - 7.Web Setup - 2.Status menu] rooter function, IP filter, Log - Added the following devices to Configurable type AIO of the [Appendix - 2.Device I/F] CPS-AO-1604ALI - Added the following devices to Configurable type DIO of the [Appendix - 2.Device I/F] CPS-DI-16L, CPS-DI-16RL, CPS-DO-16L, CPS-DO-16RL, CPS-RRY-4PCC
November 2024	Ver 1.5.0 - Added new models CPS-MGS341-DS1 CPS-MGS341G5-DS1 - Added I/O support for the following modules CPS-SSI-4C

- Every effort has been made to ensure the accuracy of the contents of this manual. If any suspicious or missing information is discovered, please contact your retailer.
- CONPROSYS is a registered trademark of CONTEC CO., LTD. Other company and product names mentioned herein are generally trademarks or registered trademarks of their respective owners.

CONTEC CO., LTD. 3-9-31, Himesato, Nishiyodogawa-ku, Osaka 555-0025, Japan

https://www.contec.com/

CONPROSYS Linux SDK User's Manual (Cross-Build Edition)

NA04973 (LYTX273) 11222024_rev7 [04072016] November 2024 Edition