How to use coprocessor service for Android

Contents

How to use coprocessor service for Android

Stable: 09.10.2019 - 13:16 / Revision: 03.10.2019 - 08:03

This article explains how to use the CoproService which is a SystemService embedded in the Android framework, allowing to use a serial connection to communicate with the Arm® Cortex®-M processor embedded in a compatible hardware board.

See STM32MPU Embedded Software for Android architecture overview to get more information on compatible hardware boards.

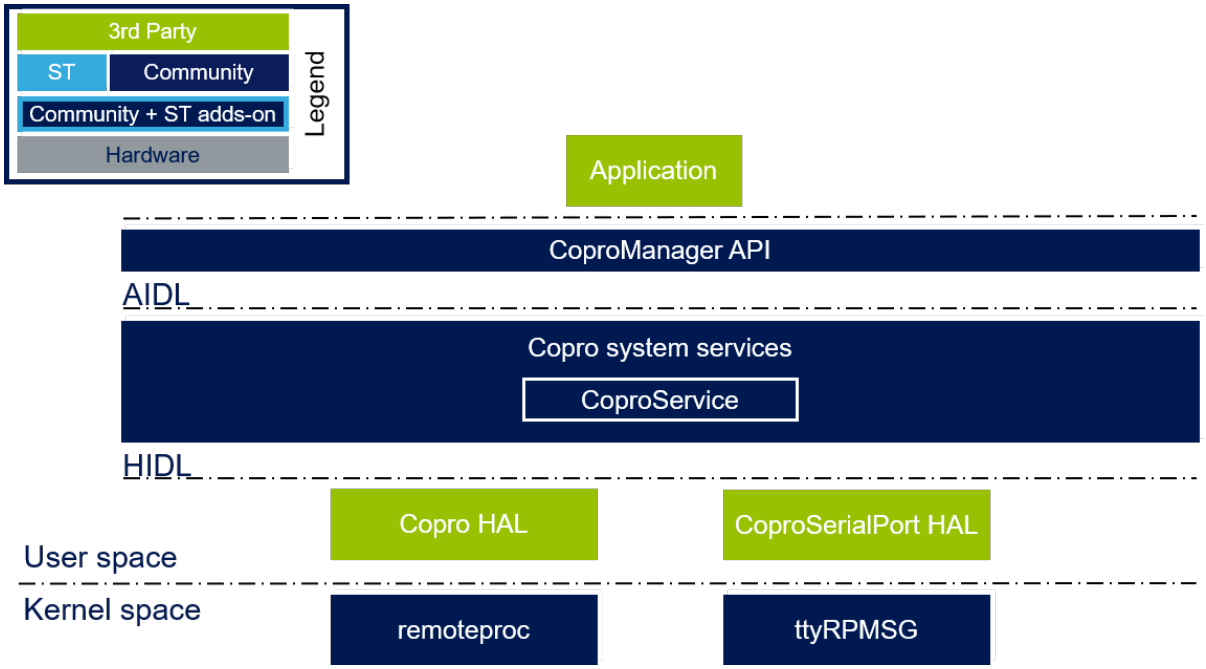
Contents

1 Prerequisites

The environment must be installed using the Developer Package adapted to your selected microprocessor device. See the list of Android Developer Package.

The firmware built for the Arm® Cortex®-M processor has to use RemoteProc and RPMSG, and has to instantiate a `/dev/ttyRPMSG` unit during its initialization.

2 CoproService



CoproService is composed of two parts: Firmware management and TTY management.

2.1 Firmware management

The Arm® Cortex®-M processor can be used to offload real time algorithms (such as MotorControl), support connectivity mediums (like LORA), etc.

The users develop their firmwares thanks to the STM32CubeMX and SW4STM32 tools.

The firmware is embedded in the Android vendor partition in the `/vendor/firmware/copro/` directory.

The CoproService looks at the contents of `/vendor/firmware/copro/` to get the list of available firmwares. It then loads the requested firmware at the user's/application's request.

2.2 TTY management

When the firmware is loaded and running, a `/dev/ttyRPMSG0` driver is available to communicate with the Arm® Cortex®-M processor.

It is up to the users to define and write their own protocol on top of `ttyRPMSG0`.

Application is then able to:

- open `ttyRPMSG0`
- write to `ttyRPMSG0`
- read from `ttyRPMSG0`

- close `ttyRPMSG0`

3 API

3.1 CoproManager API

The CoproManager API gives access to the Firmware management.

- `CoproManager getInstance()`: Gets the instance of the available `CoproManager` to start using it.
- `FirmwareInfo[] getFirmwareList()`: Returns the list of available firmwares present in `/vendor/firmware/copro/`
- `FirmwareInfo getFirmwareByName(String name)`: Looks up a firmware by its name in the `/vendor/firmware/copro/` folder and returns all useful information about it, otherwise returns `null`.
- `boolean isFirmwareRunning(int id)`: Returns `true` if the firmware of identifier `id` is running, otherwise returns `false`.
- `void startFirmware(int id)`: Starts the firmware defined by its identifier `id`.
- `void stopFirmware()`: Stops any running firmware
- `CoproSerialPort getSerialPort()`: Gets access to `CoproSerialPort` to interact with the firmware. See below for more info.

3.2 FirmwareInfo API

The FirmwareInfo API gives information about firmwares.

- `int getId()`: get the ID given by the CoproService to this firmware
- `String getName()`: get the file name of this firmware
- `boolean getState()`: returns the state of the firmware at the moment of the call: running (`true`) or stopped (`false`).

3.3 CoproSerialPort API

The CoproSerialPort API gives access to the `/dev/ttyRPMSG0` driver available when a firmware is running. It allows the application to open and close the interface and also to send and receive data.

- `void open(int mode)`: opens a file descriptor to `/dev/ttyRPMSG0` with the given mode : 0 for normal mode, 1 for raw mode
- `void close()`: closes the file descriptor to `/dev/ttyRPMSG0`
- `String read()`: reads available bytes if the file descriptor is open, otherwise an empty string is returned.
- `void write(String command)`: writes the given "command" to the file descriptor.

4 Develop an application using the coprocessor service

In order to build the application, make sure you are using the appropriate Android SDK including CoproService (refer to How to build and install an SDK for Android).

In Android Studio create a new application.

Then, do not forget to select the right API level in the application Manifest, in order to be sure that CoproService is available.

To use the CoproService you need to get the instance to the CoproManager, then you will be able to use the CoproManager API defined above:

```
CoproManager mCoproManager = CoproManager.getInstance();
```

To be able to communicate with the running firmware you need to get a CoproSerialPort from CoproManager, then you will be available to use the CoproSerialPort API defined above :

```
ICoproSerialPort mCoproSerialPort = mCoproManager.getSerialPort();
```

The serial communication is possible only if a firmware is running. Make sure to close the serial communication when you stop a firmware and open it when you start one.

TeleTYpewriter

Application programming interface

Software development kit (A programming package that enables a programmer to develop applications for a specific platform.)

Permission error

Stable: 23.06.2020 - 13:58 / Revision: 10.06.2020 - 09:24

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer

Permission error

Stable: 23.07.2020 - 14:46 / Revision: 09.07.2020 - 13:48

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer

Permission error

Stable: 04.06.2020 - 13:38 / Revision: 04.06.2020 - 13:35

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer

Permission error

Stable: 15.04.2020 - 08:59 / Revision: 15.04.2020 - 08:56

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer

Permission error

Stable: 31.01.2020 - 13:04 / Revision: 31.01.2020 - 13:02

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer

Permission error

Stable: 09.10.2019 - 13:12 / Revision: 09.10.2019 - 13:11

You do not have permission to read this page, for the following reason:

The action "Read pages" for the draft version of this page is only available for the groups ST_editors, ST_readers, Selected_editors, sysop, reviewer