Wiki > Milan Team > MCU (Zephyr) Development > 3. Running / Debugging MCU > MCU Log Collection on SOC (Revision 51505045)

MCU Log Collection on SOC

General

MCU log collection (**mcucapture** service) has been added to MOS. It comes up on boot and captures the serial port output from the MCU. It defaults to feeding the log to the SOC system log, from which it can be retrieved with logcat:

```
$ adb logcat -s mcucapture
```

Capturing to system log is subject to regular Android system log limitations - the ring buffer is finite, so data can be lost; it is also volatile across reboots.

Stopping/starting/disabling the service

You can manually stop/start the service via the following commands:

```
$ adb shell setprop mcucapture.enabled 0
$ adb shell setprop mcucapture.enabled 1
```

If you want to prevent mcucapture service from running, set the following system property:

```
$ adb shell setprop persist.vendor.mcu.capture.disabled 1
$ adb shell setprop mcucapture.enabled 0
```

Capture to Persistent Files

mcucapture service is capable of capturing logs to a set of files on the SOC, in addition to system log. Logging to files is controlled via the following MOS system properties, settable in the root shell (**adb root** must be run first). Service should be restarted per above for the settings to take effect:

Enable writing logs to files (setting is persistent across reboots; service should be restarted for it to take effect):

```
$ adb shell setprop persist.vendor.mcu.capture.log_to_file 1
```

Set maximum size of the log files, in bytes. If not set, defaults to 10MB. Setting to 0 disables size limits:

```
$ adb shell setprop persist.vendor.mcu.capture.max_log_size <size_in_bytes>
```

Set maximum number of log files to keep around. If not set, defaults to 10. Setting to 0 disables file count limit:

```
$ adb shell setprop persist.vendor.mcu.capture.max_log_cnt <count>
```

If capturing to files is enabled, log files can be found in **/data/vendor/mcucapture** (accessible via root adb pull and root shell). They will be pruned once maximum count is exceeded, oldest first.

Capture & SOC Suspend

Starting with commits per T85049990, **mcucapture** does not prevent SOC from going into doze (deep suspend) by default - e.g. if Milan ADB USB port is not connected (like when device is on the wrist), SOC will doze upon entering AOD. As a result, **mcucapture** logs will lose MCU output while SOC remains suspended. To avoid this, you can prevent dozing while **mcucapture** is running with the following persistent setting (service must be restarted for the setting to take effect):

```
$ adb shell setprop persist.vendor.mcu.capture.nodoze 1
```

Note that this comes with a power penalty and will shorten the duration of time that Milan can run on a single battery charge.

Serial Port Access

While **mcucapture** service is running, it holds a lock on /dev/ttyHS0. Programs that need to access the port (**mcu_flash.sh**, **mcu_picocom.sh**) stop mcucapture on entry and restart it on exit. As a result, while **mcu_picocom.sh** or **mcu_flash.sh** are running, capturing is disabled.

« 3. Running / Debugging MCU

Debugging MCU kernel crashes »

)}