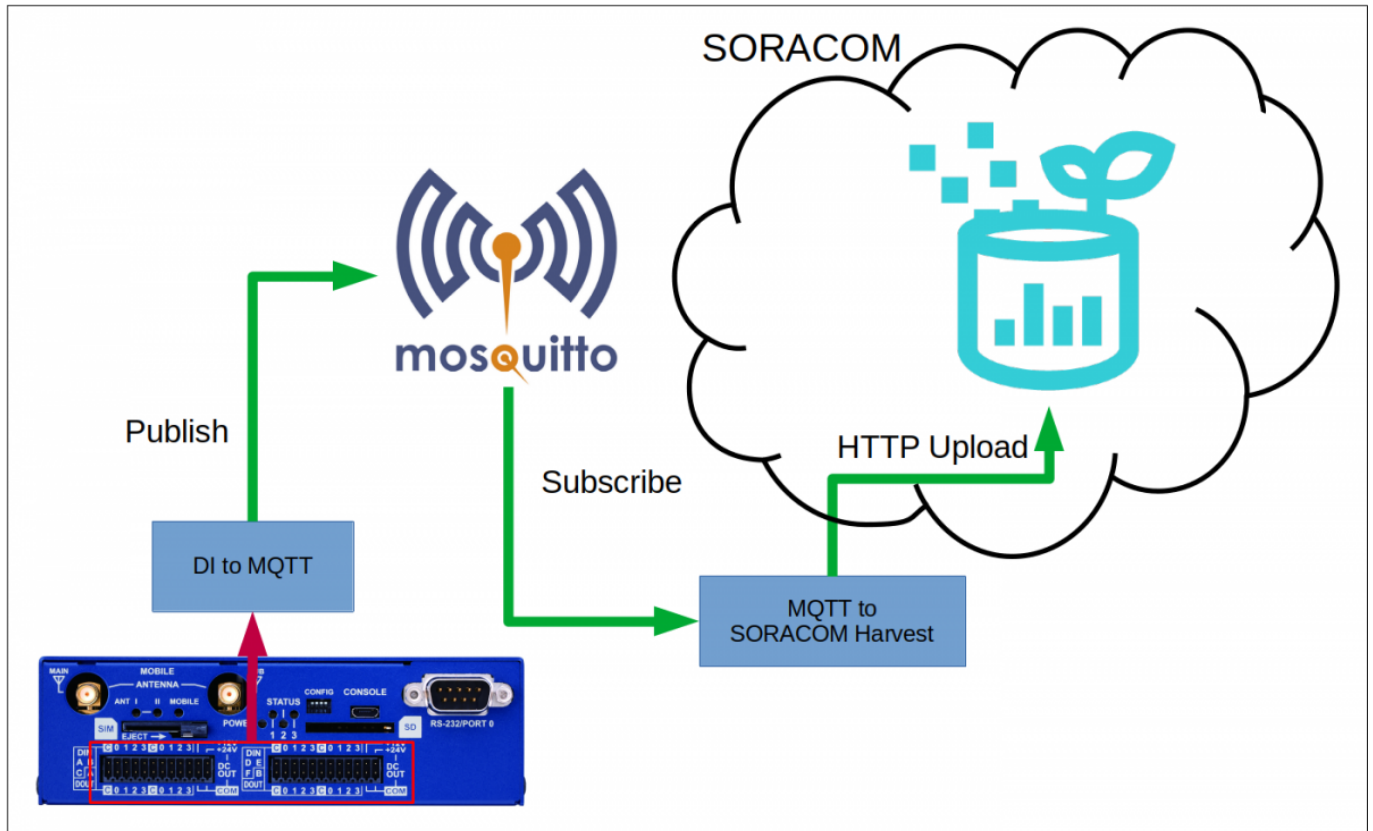


目次

DI 変化で状態を SORACOM Harvest にアップロードする	3
アプリケーションの構成	3
アプリケーションの起動	3
DI to MQTT	4
MQTT to SORACOM Harvest	4
MQTT テスト (確認用)	4
動作例	5
SORACOM コンソールで確認	5
ソースコード	6
How to Build	6
必要なソフトウェアのインストール	6
ソースコードの展開	10
ソースのビルド	10

DI 変化で状態を SORACOM Harvest にアップロードする

MA-X3xx の DI の状態を監視し、変化があったときに監視対象の DI の状態を [SORACOM Harvest](#) サービスにアップロードするサンプルです。



アプリケーションの構成

このサンプルは、下記により構成されています。

- MA-X3xx 内部の MQTT Broker(mosquitto)
- DI の状態を監視¹⁾し、DI 状態変化時に MQTT Broker に DI の状態を Publish するアプリケーション(di_mqtt)
- MQTT Broker を Subscribe し、DI の状態を受け取ったら SORACOM Harvest Data にアップロードするアプリケーション(mqtt_harvest)

アプリケーションの起動

DI to MQTT

監視したい DI のリストを定義した設定ファイルを作成します。

[di_mqtt.conf](#)

```
[MQTT]
HOST = localhost
PORT = 51883
TOPIC = DI

[DI]
DI_0 = Raw
DI_1 = Raw
DI_2 = Raw
DI_3 = Filtered
DI_4 = Raw
DI_5 = Raw
DI_6 = Raw
DI_7 = Raw
DI_8 = Raw
DI_9 = Raw
DI_10 = Raw
DI_11 = Raw
```

上記のような設定ファイルを指定して、コマンドを実行します。

```
root@metis:/tmp# ./di_mqtt -c di_mqtt.conf
```

MQTT to SORACOM Harvest

DI を監視するコマンドと同じ設定ファイルを指定して、コマンドを実行します。

```
root@metis:/tmp# ./mqtt_soracom -c di_mqtt.conf
```

MQTT テスト (確認用)

MQTT で publish されるデータを確認することもできます。

```
user1@metis:~$ nmqtt_sub -p 51883 -t DI
```

動作例

2つのコマンドを実行している状態で DI の状態を変化させると MQTT Broker に DI の状態が publish され、それを MQTT Broker 経由で受け取ったプログラムが SORACOM Harvest Data にアップロードします。

MQTT 確認用で subscribe しているプログラムの出力

```
user1@metis:~$ nmqtt_sub -p 51883 -t DI
DI:
{"timestamp":1687424585957,"state":{"DI_10":0,"DI_3":0,"DI_8":0,"DI_0":0,"DI_5":0,"DI_4":0,"DI_6":0,"DI_9":0,"DI_11":0,"DI_7":0,"DI_1":0,"DI_2":0}}
```

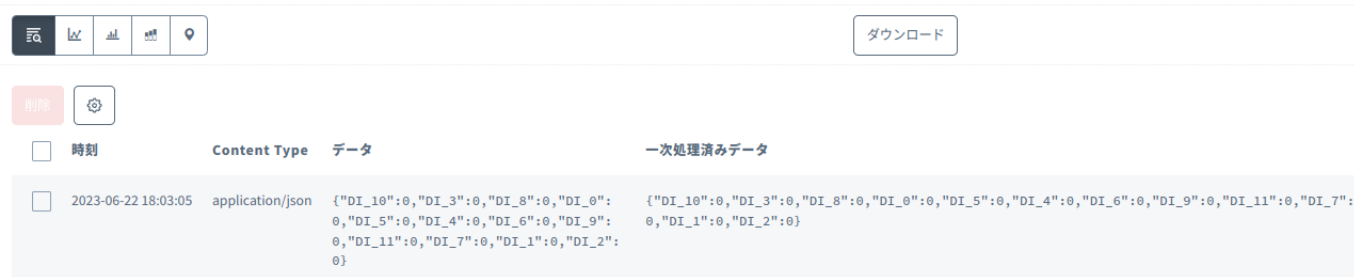
syslog に SORACOM Harvest Data へアップロードした、というログも記録されます。

```
Jun 22 18:03:06 metis mqtt_soracom[8021]: upload: timestamp: 1687424585957
Jun 22 18:03:06 metis mqtt_soracom[8021]: upload: payload:
{"DI_10":0,"DI_3":0,"DI_8":0,"DI_0":0,"DI_5":0,"DI_4":0,"DI_6":0,"DI_9":0,"DI_11":0,"DI_7":0,"DI_1":0,"DI_2":0}
Jun 22 18:03:06 metis mqtt_soracom[8021]: upload: response code: 201 Created
```

※ **“timestamp”** は SORACOM Harvest Data の timestamp に設定するため、データから抜き出して HTTP Header **“X-Soracom-timestamp”** にセットするようになっています。

SORACOM コンソールで確認

アップロードされたデータを SORACOM コンソール で確認してみます。



The screenshot shows the SORACOM Harvest Data console interface. At the top, there are navigation icons and a 'ダウンロード' (Download) button. Below that, there are '削除' (Delete) and '設定' (Settings) buttons. The main area displays a table with columns for '時刻' (Time), 'Content Type', 'データ' (Data), and '一次処理済みデータ' (Once-processed data). A single data entry is visible, corresponding to the example data provided in the text.

時刻	Content Type	データ	一次処理済みデータ
2023-06-22 18:03:05	application/json	{"DI_10":0,"DI_3":0,"DI_8":0,"DI_0":0,"DI_5":0,"DI_4":0,"DI_6":0,"DI_9":0,"DI_11":0,"DI_7":0,"DI_1":0,"DI_2":0}	{"DI_10":0,"DI_3":0,"DI_8":0,"DI_0":0,"DI_5":0,"DI_4":0,"DI_6":0,"DI_9":0,"DI_11":0,"DI_7":0,"DI_1":0,"DI_2":0}

きちんとアップロードされていることが確認できました。

ソースコード

サンプルアプリのコードです。

File	Stat	SHA1SUM	Info
di_mqtt.tar.xz	2023/06/22 19:23 8.5 KB	f6d915c19bfcac942885c36ebe635e8cb455ae04	DI to MQTT/MQTT to SORACOM Harvest
nim_gpiod.tar.xz	2023/06/22 19:23 3.1 KB	97eb2b65b748f4bd049a9a7e60ff157a88aa5127	libgpiod wrapper library

How to Build

アプリケーションのビルド方法です。

必要なソフトウェアのインストール

C Compiler (build-essential)

[ファームウェアの作成](#) でも必要なのでインストールしておきます。

参照: [開発環境のセットアップ\(クロス\[arm64 用\]\)](#)

Nim Compiler

[Nim 言語](#) で記述しているので、[Nim Compiler をインストール](#) します。

```
user1@max3xx-devel:~$ curl https://nim-lang.org/choosenim/init.sh -sSf | sh
choosenim-init: Downloading choosenim-0.8.4_linux_amd64
  Prompt: Can choosenim record and send anonymised telemetry data? [y/n]
  ... Anonymous aggregate user analytics allow us to prioritise
  ... fixes and features based on how, where and when people use Nim.
  ... For more details see: https://goo.gl/NzUEPf.
  Answer: n
Downloading Nim 1.6.12 from nim-lang.org
[#####] 100.0% 0kb/s
Extracting nim-1.6.12-linux_x64.tar.xz
Extracting nim-1.6.12-linux_x64.tar
Building Nim 1.6.12
```

```
Compiler: Already built
Installed component 'nim'
Installed component 'nimble'
Installed component 'nimgrep'
Installed component 'nimpretty'
Installed component 'nimsuggest'
Installed component 'testament'
Installed component 'nim-gdb'
Switched to Nim 1.6.12
choosenim-init: ChooseNim installed in /home/user1/.nimble/bin
choosenim-init: You must now ensure that the Nimble bin dir is in your PATH.
choosenim-init: Place the following line in the ~/.profile or ~/.bashrc
file.
choosenim-init:     export PATH=/home/user1/.nimble/bin:$PATH
user1@max3xx-devel:~$
```

最後の部分で指示されているようにPATHを設定しておきます。

```
user1@max3xx-devel:~$ nano -w .bashrc
```

```
ファイル(F) 編集(E) 表示(V) 検索(S) 端末(T) ヘルプ(H)
GNU nano 6.2 .bashrc *
case "$TERM" in
xterm*|rxvt*)
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@h: \w\a]$PS1"
    ;;
*)
    ;;
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "[[ $? = 0 ] && echo terminal || echo error]" "$(history|tail -n1|sed -e'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

export PATH=/home/user1/.nimble/bin:$PATH

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line  M-E Redo
```

最後の行に追加しておきます。

実行できるように .bashrc を読み直し、実行の確認をしてみます。

```
user1@max3xx-devel:~$ source .bashrc
user1@max3xx-devel:~$ nim -v
Nim Compiler Version 1.6.12 [Linux: amd64]
Compiled at 2023-03-10
Copyright (c) 2006-2023 by Andreas Rumpf
```



```
git hash: 1aa9273640c0c51486cf3a7b67282fe58f360e91
active boot switches: -d:release
user1@max3xx-devel:~$
```

きちんとインストールされて実行できることが確認できました。

Nim Package List 更新

Nim のパッケージリストを更新しておきます。

```
user1@max3xx-devel:~$ nimble refresh
Downloading Official package list
  Success Package list downloaded.
user1@max3xx-devel:~$
```

libgpiod 開発用(-dev)パッケージの追加

apt で追加しておきます。

```
user1@max3xx-devel:~$ sudo apt install libgpiod-dev
[sudo] password for user1:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpiod2
Suggested packages:
  libgpiod-doc
The following NEW packages will be installed:
  libgpiod-dev libgpiod2
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 103 kB of archives.
After this operation, 492 kB of additional disk space will be used.
Do you want to continue? [Y/n]
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libgpiod2 amd64
1.6.3-1build1 [42.2 kB]
Get:2 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libgpiod-dev
amd64 1.6.3-1build1 [61.1 kB]
Fetched 103 kB in 2s (52.0 kB/s)
Selecting previously unselected package libgpiod2:amd64.
(Reading database ... 33935 files and directories currently installed.)
Preparing to unpack .../libgpiod2_1.6.3-1build1_amd64.deb ...
Unpacking libgpiod2:amd64 (1.6.3-1build1) ...
Selecting previously unselected package libgpiod-dev:amd64.
```

```
Preparing to unpack .../libgpiod-dev_1.6.3-1build1_amd64.deb ...
Unpacking libgpiod-dev:amd64 (1.6.3-1build1) ...
Setting up libgpiod2:amd64 (1.6.3-1build1) ...
Setting up libgpiod-dev:amd64 (1.6.3-1build1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.1) ...
user1@max3xx-devel:~$
```

ソースコードの展開

適当なディレクトリ以下に、2つのソースのアーカイブを展開します。

```
user1@max3xx-devel:~$ cd src/
user1@max3xx-devel:~/src$
```

```
user1@max3xx-devel:~/src$ tar axf ../nim_gpiod.tar.xz
user1@max3xx-devel:~/src$ tar axf ../di_mqtt.tar.xz
user1@max3xx-devel:~/src$ ls -l
total 12
drwxrwxr-x  3 user1 user1 4096 Jun 22 19:22 di_mqtt  <----
drwxrwxr-x 27 user1 user1 4096 Apr 13 13:50 linux-kernel
drwxrwxr-x  3 user1 user1 4096 Jun 22 19:22 nim_gpiod <----
user1@max3xx-devel:~/src$
```

ソースのビルド

nim_gpiod のインストール

di_mqtt のソースをビルドするために必要なので[nim_gpiod²⁾をインストールします。

```
user1@max3xx-devel:~$ cd src/nim_gpiod/
user1@max3xx-devel:~/src/nim_gpiod$ nimble install --cpu:arm64
Verifying dependencies for nim_gpiod@0.2.0
Installing nim_gpiod@0.2.0
Building nim_gpiod/nim_gpiod using c backend
Success: nim_gpiod installed successfully.
user1@max3xx-devel:~/src/nim_gpiod$
```

di_mqtt のビルド

ビルドする準備ができたので di_mqtt をビルドします。

```
user1@max3xx-devel:~/src/nim_gpiod$ cd ../di_mqtt/
user1@max3xx-devel:~/src/di_mqtt$ nimble build -d:release --cpu:arm64
  Verifying dependencies for di_mqtt@0.2.0
  Installing argparse@0.10.1
Downloading https://github.com/iffy/nim-argparse using git
  Verifying dependencies for argparse@0.10.1
  Installing argparse@0.10.1
  Success: argparse installed successfully.
  Installing nmqtt@>= 1.0.5
Downloading https://github.com/zevv/nmqtt using git
  Verifying dependencies for nmqtt@1.0.5
  Installing cligen@>= 0.9.45
Downloading https://github.com/c-blake/cligen.git using git
  Verifying dependencies for cligen@1.6.6
  Installing cligen@1.6.6
  Success: cligen installed successfully.
  Installing bcrypt@>= 0.2.1
Downloading https://github.com/ithkuil/bcryptnim using git
  Verifying dependencies for bcrypt@0.2.1
  Installing bcrypt@0.2.1
  Success: bcrypt installed successfully.
  Installing nmqtt@1.0.5
  Building nmqtt/nmqtt/nmqtt_password using c backend
  Building nmqtt/nmqtt/nmqtt using c backend
/tmp/nimble_22433/githubcom_zevvnmqtt_1.0.5/nmqtt.nim(285, 54)
template/generic instantiation of `async` from here
/tmp/nimble_22433/githubcom_zevvnmqtt_1.0.5/nmqtt.nim(292, 5) Warning: The
bare except clause is deprecated; use `except CatchableError:` instead
[BareExcept]
/tmp/nimble_22433/githubcom_zevvnmqtt_1.0.5/nmqtt.nim(285, 54)
template/generic instantiation of `async` from here
/home/user1/.choosenim/toolchains/nim-1.6.12/lib/pure/asyncmacro.nim(42, 5)
Warning: The bare except clause is deprecated; use `except CatchableError:`
instead [BareExcept]
... 中略 ...
/home/user1/.choosenim/toolchains/nim-1.6.12/lib/pure/asyncmacro.nim(42, 5)
Warning: The bare except clause is deprecated; use `except CatchableError:`
instead [BareExcept]
/tmp/nimble_22433/githubcom_zevvnmqtt_1.0.5/nmqtt/nmqtt_sub.nim(20, 222)
template/generic instantiation of `async` from here
/home/user1/.choosenim/toolchains/nim-1.6.12/lib/pure/asyncmacro.nim(42, 5)
Warning: The bare except clause is deprecated; use `except CatchableError:`
instead [BareExcept]
  Success: nmqtt installed successfully.
  Info: Dependency on nim_gpiod@>= 0.2.0 already satisfied
  Verifying dependencies for nim_gpiod@0.2.0
  Installing jsony@>= 1.1.5
Downloading https://github.com/treeform/jsony using git
  Verifying dependencies for jsony@1.1.5
```

```
Installing jsony@1.1.5
  Success: jsony installed successfully.
  Building di_mqtt/di_mqtt using c backend
  Building di_mqtt/mqtt_soracom using c backend
user1@max3xx-devel:~/src/di_mqtt$
```

必要なライブラリが芋づる式に追加され、di_mqtt のビルドが完了しました。

できあがったバイナリを確認してみます。

```
user1@max3xx-devel:~/src/di_mqtt$ ls -l bin/
total 1272
-rwxrwxr-x 1 user1 user1 563040 Jun 22 19:33 di_mqtt
-rwxrwxr-x 1 user1 user1 733408 Jun 22 19:33 mqtt_soracom
user1@max3xx-devel:~/src/di_mqtt$
```

strip によりシンボルを除去してバイナリを小さくすることもできます。

```
user1@max3xx-devel:~/src/di_mqtt$ aarch64-linux-gnu-strip bin/*
user1@max3xx-devel:~/src/di_mqtt$ ls -l bin/
total 952
-rwxrwxr-x 1 user1 user1 423936 Jun 22 19:35 di_mqtt
-rwxrwxr-x 1 user1 user1 546816 Jun 22 19:35 mqtt_soracom
user1@max3xx-devel:~/src/di_mqtt$
```

1)

libgpiod を使用しています

2)

libgpiod の Nim 用ラッパーライブラリ

From: <https://ma-tech.centurysys.jp/> - MA-X/MA-S/MA-E/IP-K Developers' Wiki

Permanent link: https://ma-tech.centurysys.jp/doku.php?id=max3xx_tips:upload_di_state_to_harvest:start

Last update: **2023/06/23 08:56**

