

# Application Note : No.20230613-00.00

## External **ARMM**

改版履歴:	2
1. 概要	2
a. 本App noteの目的	3
b. 対象者	3
2. ARMMの特徴	3
a. 超低消費電力動作	3
b. スーパーキャパシタ搭載	3
c. DC12V動作	4
d. ケース	4
e. Edge AI BoxとはUSBケーブル接続	4
Edge AI Box とはUSBケーブルにて接続し、シリアル通信を行います。	4
f. 電源制御時間設定SW	4
3. 詳細説明	5
a. ARMMの利用目的	5
b. AI Boxとの接続図	5
c. AI BoxとARMM間のシリアル通信	6
i. シリアル通信設定	6
ii. 通信パケット仕様	6
iii. コマンド一覧	7
d. AI Box側のプログラム例	7
e. Log File	7
4. 利用手順	8
5. 質問等対応	9

### 改版履歴:

Apr. 23, 2024    初版

## 1. 概要

ARMM(Auto Recovery Monitoring Module)はEDGEMATRIX Inc.が開発した32ビットMCUを搭載した超小型ボードです。Edge AI Box(以後単にAI Boxと呼ぶ)とシリアル通信を介してheart beatや各種コマンドのやりとりを行います。またAI Boxの主電源をON/OFFできる機能を持っており、AI Boxからのheart beat途絶を検出した場合にAI Boxの主電源を入れ直し、AI Boxの回復を試みます。

AI Boxが設置された環境の変化や予期せぬAI処理の異常、見落とされたバグ、電源の瞬断、落雷によるサージ大電流、高速AI処理などによる発熱、モバイルブロードバンド通信異常など、AI Boxが異常動作となる要因は数多くあります。異常動作時には予想できなかった動作や動作停止が発生し、運よくそれらを検出してソフトウェアによるリブート(ウォームブート)を行ったとしても復帰できない場合も発生します。そのような場合、多くは現地に行っておもとの電源(主電源)を一度切り、放電を待ったのち再投入を行い(コールドブート)、AI Boxの正常動作復帰を確認します。特に遠隔地の場合、現地への移動時間や費用の負担がおおきくなります。また停止中はAI処理が行われないので速やかに復帰することが期待されます。

もちろん市場にはこのような機能を持った機器が販売されています。つまり、ネットワーク経由などで対象機器を監視して、動作異常を検出した場合、ウォームブートやコールドブートを行わせることができます。しかしながら、ネットワーク機能が正常に動作していることを前提とした比較的高価な機器やクラウドソリューションになっており、機器スペースやランニング費用がかかります。また電源を分離しなければならないなど、電源周りの設計や格納スペースが必要です。

ARMMは、その様な問題点を考慮し、必要最低限の機能と容積で、安価で十分なりカバリ機能をAI Boxに提供できるように設計されました。ARMMでAI Boxは異常状態から短時間で自動復帰し、安定してAI動作を続けることができ、保守費を削減することができます。さらに付加機能として、主電源の停止記録、シリアル通信を使ったNVIDIA® Jetson™との通信を介したheart beat 記録や各種コマンドの実行を行うことができます。

ARMMはEdge AI Box本体側のARMMアプリケーションと連携して動作することを前提としています。本体側のARMMアプリケーションについては以下を参照ください。

<https://github.com/hsugihara/ARMM>

ARMMには外付け型と内蔵型の2種類があります。本アプリケーションノートではExternal ARMM(外付け型ARMM)について説明します。

なお、Internal ARMM(内蔵型ARMM)ではNVIDIA® Jetson™側で実行されるプログラムにて、モバイルブロードバンドモジュールのみの電源OFF/ON、PoE電源のみのOFF/ONなど、異常発生頻度の高い部分の電源のみをOFF/ONできる機能を追加し、NVIDIA® Jetson™CPU動作を止めることなく、短時間で正常復帰を行える機能などが

追加されています。Internal ARMMに関してはAN20240109 Internal ARMM アプリケーションノートをご参照ください。

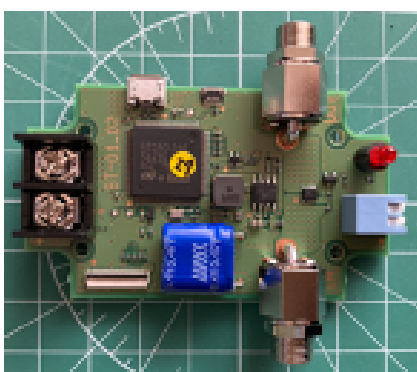
#### a. 本App noteの目的

本アプリケーションノートはExternal ARMM(外付け型ARMM、以下ARMMと呼ぶ) を利用するユーザー向け説明書です。

#### b. 対象者

HW設計、STM32 MCU、シリアル通信、Pythonプログラム、Ubuntuコマンドやシェルスクリプトなどの知識をもつエンジニアを対象とします。

## 2. ARMMの特徴

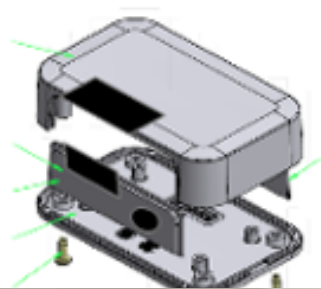


#### a. 超低消費電力動作

最大消費電力75mWattと超消費電力です。動作時実測では11 mA x 3.3Volt = 36.3mWattでした。

#### b. スーパーキャパシタ搭載

主電源停止時には電源停止を記録し、待機状態になり、電源回復を待ちます。待機時には10-30 $\mu$ Aしか消費しないため、1日以上待機時間があります。



#### c. DC12V動作

Edge AI Box LightのACアダプターを利用可能とし、Light使用時には1個のACアダプターにてARMMもLightも動作します。



#### d. ケース

TAKACHI PF8-2-5Dケースに収まるサイズになります。ただしサイドパネルを取り去るか、穴を開ける必要があります。

#### e. Edge AI BoxとはUSBケーブル接続

Edge AI Box とはUSBケーブルにて接続し、シリアル通信を行います。

#### f. 電源制御時間設定SW

DSW1とDSW2の組合せでEdge AI Boxからrebootコマンドを受けたのちの電源OFFまでの時間と、その後のONするまでの時間を設定できます。以下を参照ください。

DSW1 : OFFするまでの時間 (linux shutdown時間)

L: 120秒 (EDGEMATRIX Service向け設定)

H: 40秒 (通常利用時)

DSW2 : ONするまでの時間 (Edge AI Box 放電時間)

L: 60秒 (outdoor機)

H: 10秒 (indoor機)

### 3. 詳細説明

#### a. ARMMの利用目的

ARMMを利用することで、基本的に次の2項目を実行できます。

1. 思わぬ障害などにより、AI Boxが停止した場合、AI Boxの電源を入れ直すことができる。
2. 主電源が喪失した場合の時刻を特定できる。

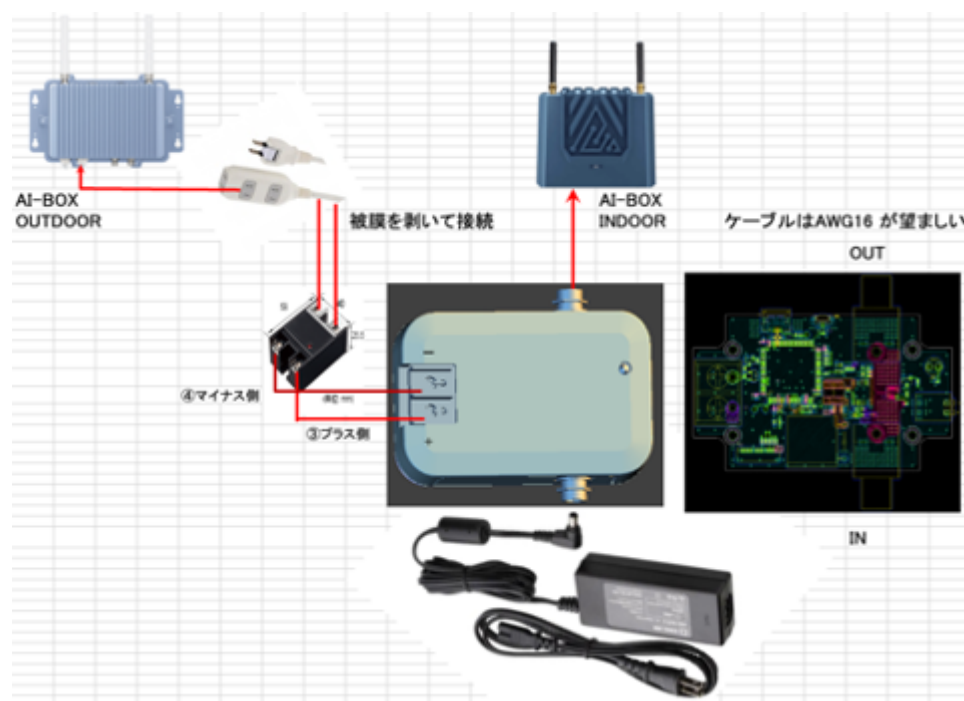
## b. AI Boxとの接続図

以下に示すように、ARMMはAI Box Lightに使用しているACアダプターから12DCVを受けて動作します。

AI Box LightにはARMMから12DCVを直接供給することができ、かつその12DCVを供給/停止することができます。

AI Box Standard Outdoorの場合には100ACV入力をSSRにて制御することで供給/停止することができます。

また、各AI BoxからUSBケーブル(USB type A to micro)にてARMMに接続し、シリアル通信を行います。



## c. AI BoxとARMM間のシリアル通信

### i. シリアル通信設定

	外付け型ARMM
通信ポート	USB2.0 (COM)
キャリアボー	USB2.0 ケーブル

ドとの接続	
通信フォーマットとボーレート	Stop bit : 1 Parity bit : non 115200bps

ii. 通信パッケージ仕様

<DLE><STX>

[payload

[command 1 byte]

[parameter 0~1024 bytes]

[checksum 1 byte]

]

<DLE><ETX>

### iii. コマンド一覧

#### コマンド一覧

コマンド名	コマンド値	通信方向 A=AIBOX B=BT-01	用途
alive_req	0x55	A → B	AIBOX 稼働中定期的に BT-01 に対して通知する。通知が一定期間途絶えた場合 BT-01 は AIBOX をリスタートする。
alive_res	0xAA	A ← B	alive_req を受け取った際の BT-01 からの応答。
status_req	0x01	A → B	BT-01 のステータスを問い合わせる。
status_res	0x81	A ← B	BT-01 のステータスを返す。
time_sync_req	0x02	A → B	BT-01 に対して日時を設定する。
time_sync_res	0x82	A ← B	日時設定に対する応答。BT-01 に設定した日時を返す。
log_req	0x03	A → B	ログを 1 つ要求する。
log_res	0x83	A ← B	最古のログを 1 つ送る。ログが無い場合は“NO LOG”を返す
reboot_req	0x04	A → B	AIBOX のリポートを要求する。一定時間後に BT-01 は AIBOX をコールドブートする。
reboot_res	0x84	A ← B	リポート要求に対する応答。
nop_cmd	0x00	A → B	特に意味はないが送信した場合に使用。
unknown_res	0xFF	A ← B	不明なコマンドを受け取った場合の応答。

#### d. AI Box側のプログラム例

<https://github.com/hsugihara/ARMM>

#### e. Log File

ARMM本体側ではBT-logファイルにlogを書き出します。ファイルは毎日AM2:15頃、更新され、古いBT-logはBT-log.1、BT-log.1はBT-log.2と、BT-log.7までシフトされます。

log読み出しでは、ARMMボードからlogを読み出し、BT-logファイルへ書き込みます。

フォーマットは以下になります。

log format :
ログ種別(5bytes) ","(1byte) 日時分(19bytes) ","(1byte) ログ内容(36bytes) blank(2bytes)
ログ種別は下記の3通り(5bytes)
1. BOOT
2. ERROR
3. INFO
ログ保存トリガ (4トリガ)
1.BT-01起動時 初回スタンダロンモードに入った際のログを残す BOOT ,2000/01/01 00:00:00,BT-11 BOOTED
2.モード遷移時 モード遷移したときに遷移したモードをINFOで残す INFO ,YYYY/MM/DD hh:mm:ss,STAND ALONE MODE INFO ,YYYY/MM/DD hh:mm:ss,AIBOX MONITOR MODE INFO ,YYYY/MM/DD hh:mm:ss,REBOOT MODE INFO ,YYYY/MM/DD hh:mm:ss,POWER SAVING MODE
3.エラー発生時 プログラムのエラーが発生した際にERROR (未定義)
4.AIBOX OFF AIBOXの電源を落とした際にBOOT BOOT ,YYYY/MM/DD hh:mm:ss, VDET POWER DOWN

## 4. 利用手順

- a. EX3あるいはEX5側に通信制御用プログラムをインストールおよび自動起動設定をおこないます。



b. 外付けARMMを接続し、電源を立ち上げます。

## 5. 質問等対応

a. 質問は [device\\_support@edgematrix.com](mailto:device_support@edgematrix.com) にて受け付けております。