

Application Note

MSPM0 低消費電力マイコンでエッジ AI を使用した AC アーク故障検出



Akshat Aggarwal, Laavanaya Dhawan, Nathan Nohr, Vinamra Shrivastava

概要

このアプリケーション ノートでは、エッジ AI を使用して、ニューラル プロセッシング ユニット (NPU) を内蔵した低コストの MSPM0G5187 マイコン上で、リアルタイム AC アーク故障検出機能を実装する方法を紹介します。この設計では、TIDA-010971 アナログ フロント エンドのリファレンス デザインと機械学習推論を組み合わせることにより、住宅用および商用の電気システムでの直列アーク故障を検出します。目標性能には、UL 1699 ガイドラインに従ってマスキング 負荷への耐性を維持しながら、10ms 未満の応答時間と 95% を超える検出精度があります。このリファレンス デザインを採用すると、NEC (National Electrical Code: 米国電気工事規程) の要件に準拠するアーク故障回路遮断器 (AFCI) 製品を開発できます。

商標

すべての商標は、それぞれの所有者に帰属します。

目次

1 概要	2
2 クイック スタート ガイド	6
3 技術的な背景	9
4 システム要件	13
5 ハードウェア設計	15
6 モデル開発	16
7 ファームウェアの実装	22
8 サンプル コード	24
9 性能ベンチマーク	25
10 消費電力の最適化	25
11 テストと検証	25
12 トラブルシューティング	26
13 まとめ	27
14 参考資料	28
15 用語	28
16 よくある質問	29

1 概要

アーク故障の問題

アーク故障は、住宅や商業ビルの電気火災の主な原因の 1 つです。アーク故障は、電流が意図しない経路を流れ、周囲の物質を発火させる可能性のある高温のプラズマ アークを生じさせることで発生します。米国消費者製品安全委員会 (CPSC) と全米防火協会 (NFPA) によると、アーク故障は、年間 30,000 件を超える家庭火災を引き起こし、その結果数百人が死亡し、十億ドルを超える物的損害が発生しています。(出典: NFPA 電気火災統計、2019 ~ 2023 年)

表 1-1. アーク故障のタイプ

タイプ	説明	特性
直列アーク	負荷と直列で高インピーダンス接続 (配線の損傷、接続の緩みなど)	電流がアークを流れる。負荷は働くが、電力は減少
並列アーク	絶縁の損傷による、ライン - ニュートラル間、またはライン - グランド間の故障	高い故障電流。負荷を接続しなくても発生する可能性がある

直列アーク故障と並列アーク故障の比較。直列アークは、回路内の負荷により電流パス内で発生し、並列アークはラインとニュートラルまたはグラウンド間で発生します

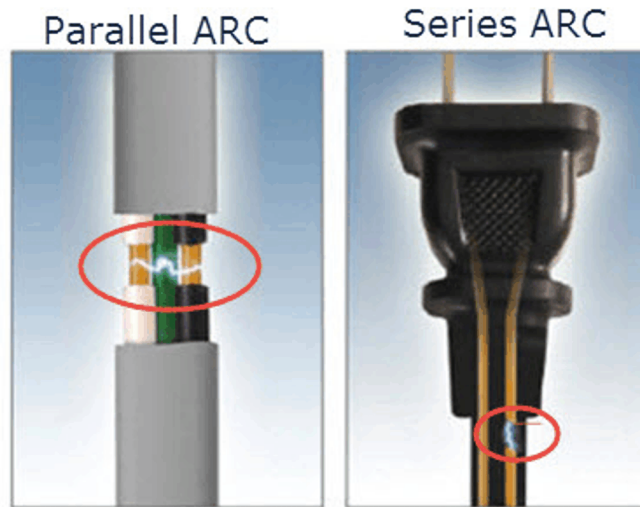


図 1-1. 直列および並列アーク故障

従来の保護が失敗する理由

従来型のサーキットブレーカおよびヒューズは、過電流状態から保護しますが、以下の理由により、アーク故障を検出できません。

- 直列アークは、正常な負荷よりも流れる電流が少ない場合がある
- アーク電流波形は、シンプルな過電流シグネチャとは異なる
- アーク放電の周波数 (kHz ~ MHz) は、サーマル ブレーカでは監視されない

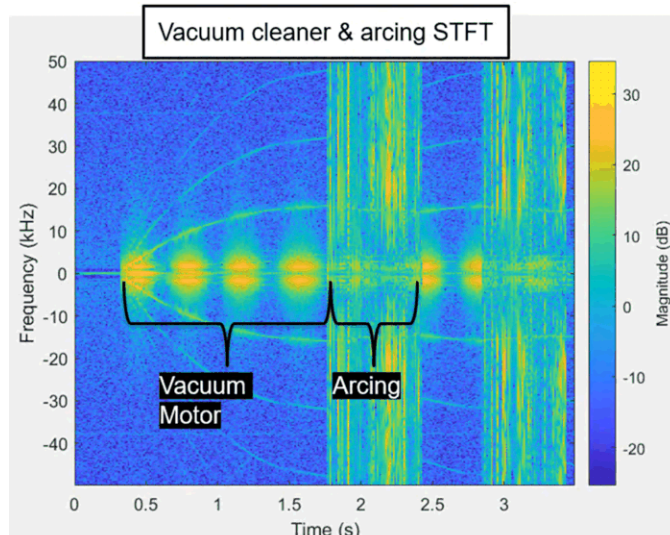


図 1-2. アーク故障と正常な通常負荷波形との関係

UL 1699 規格の概要

UL 1699 は、住宅用回路保護デバイスのテスト要件を定義した、アーク故障回路遮断器 (AFCI) 用の安全規格です。北米で販売される AFCI には UL 1699 への準拠が必要であり、米国電気法 (NEC) はほとんどの居住空間での AFCI の保護を義務付けています。

表 1-2. 主な UL 1699 テスト カテゴリ

テスト カテゴリ	説明	ML モデル関連
炭化経路アーク	アークが炭化絶縁体を通過	プライマリ検出ターゲット
点接触アーク	金属と金属の断続的な接触	高周波シグネチャ検出
マスク テスト	アークシグネチャを模倣した正常負荷	誤検出の防止
誤切断	調光器、掃除機、ドリル、SMPS	これらについてモデルの誤切断がないこと

このアプリケーションノートでは、エンジニアによる AFCI 製品の開発に役立つリファレンス デザインを示します。量産デバイスには、正式な UL 認証テストが必要です。ここに示す設計原理とテスト手法は、UL 1699 の要件と整合しています。

表 1-3. 設計目標と UL 1699 の関係

メトリック	UL 1699 の要件	この設計目標
直列アーク検出	必須	あり
並列アーク検出	必須	なし
応答時間	テストによって変化	<10ms
マスキング負荷耐性	誤切断がないこと	99% 以上の特異性

アーク故障検出にエッジ AI を使用する理由

従来のアーク故障検出では、アナログ回路を使用して、特定の周波数帯域や波形特性を監視します。これらのアプローチは効果的ではありますが、次のような制約もあります。

- 固定スレッシュホールドでは、さまざまな負荷条件に対応できない
- 限られた特徴抽出では、複雑なアーク シグネチャが得られない
- 負荷条件が厳しく、誤検出率が高い (調光器、モーター)

エッジ AI では、根本的に異なるアプローチを利用できます。

1. 複合パターン認識:ニューラル ネットワークを使用すると、アーク シグネチャと正常の負荷ノイズの間のわずかな違いを学習できます
2. 多機能分析:時間ドメイン、周波数ドメイン、エネルギー機能を組み合わせることができます
3. 適応検出:多様なデータセットでトレーニングされたモデルは、固定ルールよりも高度な普遍化が可能です
4. オンデバイス推論:クラウド接続を使用しないリアルタイム検出

表 1-4. エッジ AI と従来型アナログ AFCI の比較

要素	従来型のアナログ AFCI	エッジ AI AFCI (この設計)
検出方式	固定アナログ フィルタ + コンパレータ	学習機能を備えた ML モデル
特徴抽出	ハードウェア フィルタ (制限帯域)	ソフトウェア FFT (フル スペクトル解析)
スレッシュホールドの調整	固定または手動調整	トレーニング データから学習
誤検出の処理	追加のアナログ フィルタ	マスキング負荷に関するトレーニングされた ML モデル
更新機能	ハードウェアの変更が必要	ファームウェア / モデルの更新が可能
開発工数	アナログ回路設計の専門知識	ML トレーニング + 組み込みの導入
BOM コスト	複数のアナログ IC	NPU 搭載のシングル マイコン
消費電力	変化 (通常は高く)	低消費電力マイコン
フレキシビリティ	特定用途向け	さまざまなシナリオに適応

MSPM0 を選択する理由

MSPM0G5187 は、エッジ AI アーク故障検出に特に適しています。

表 1-5. MSPM0G5187 の特長

特長	アーク検出の利点
TinyEngine NPU	ハードウェア アクセラレーションされたニューラル ネットワーク推論:
80MHz の Cortex-M0+	リアルタイム機能抽出に適した高速 DSP
12 ビット、1.6Msps ADC	高速サンプリングはアーク シグネチャをキャプチャ
128KB フラッシュ	ML モデルおよびアプリケーション コードに十分
32KB SRAM	推論のための適切なテンソル アリナ
低消費電力	低消費電力マイコンによる常時オン監視
優れたコスト効率	コンシューマ向け AFCI 製品に適した低 BOM コスト
小型パッケージ	コンパクトな設計の 4mm×4mm オプション

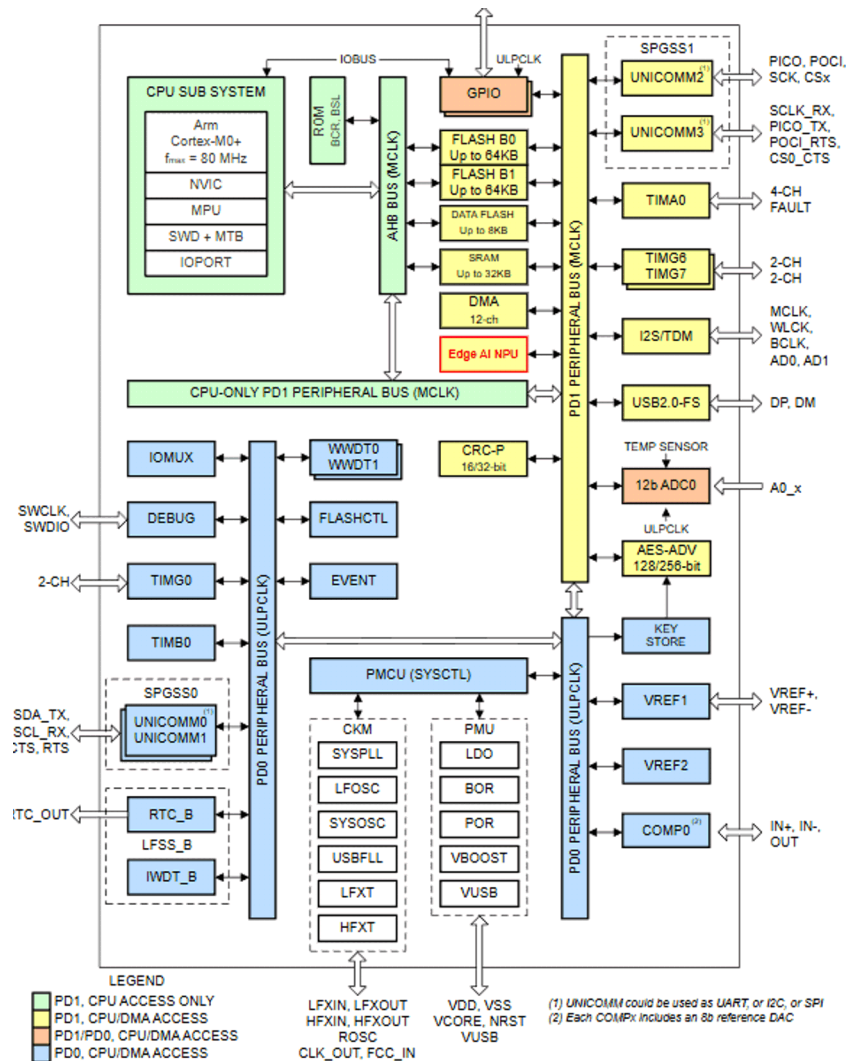


図 1-3. MSPM0G5187 の機能ブロック図

注

NPU なしの他の MSPM0 バリエーションは、ソフトウェアのみの実装を使用し、ハードウェア アクセラレーションを使用せずにエッジ AI 推論を実行することができます。

2 クイック スタート ガイド

このセクションでは、アーク故障検出のデモを実行するための方法を簡単に説明します。詳細については、後続のセクションを参照してください。

要件

必要なもの

ハードウェア:

- [LP-MSPM0G5187 LaunchPad](#)
- [TIDA-010971 アナログ フロント エンド ボード](#)
- USB Type-C ケーブル
- USB アイソレータ (AC テストに必要)

ソフトウェア:

- [Code Composer Studio 12.x 以降](#)
- [MSPM0 SDK 2.08.00 以降](#)

クイック スタート手順

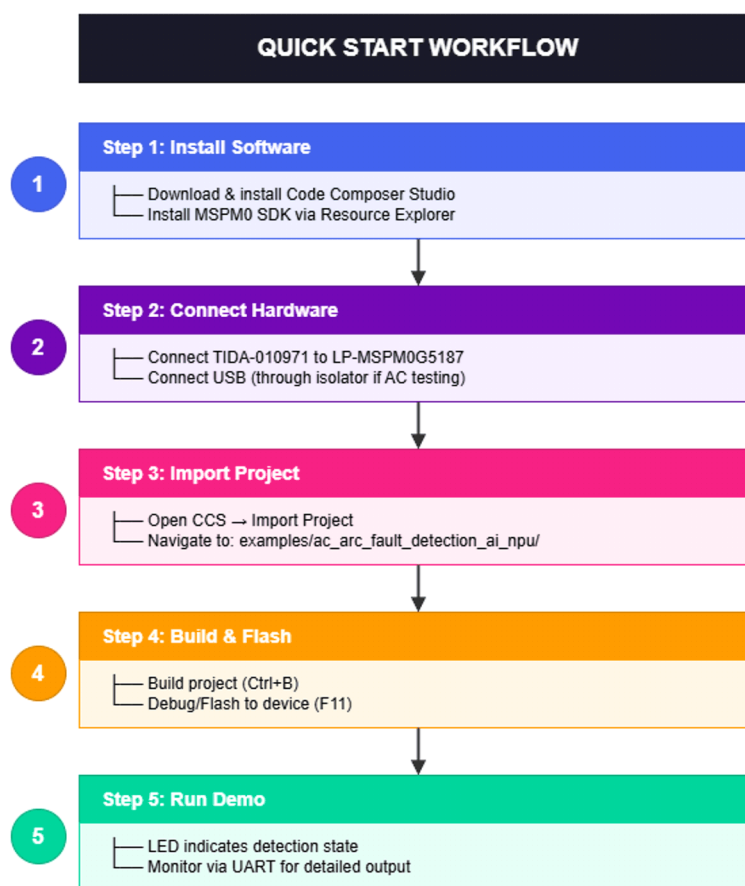


図 2-1. ソフトウェアのインストールからデモの実行までの 5 つの主なステップを示すクイック スタートのワークフロー

ステップバイステップの説明

ステップ 1: ソフトウェアをインストールする

- ti.com/ccs から Code Composer Studio をダウンロードする
- インストール時に、「MSPM0 Microcontrollers」を選択する
- CCS を起動し、「Resource Explorer」を開く (「View」→「Resource Explorer」を選択)
- 「MSPM0 SDK」を検索し、バージョン 2.10.00 以降をインストールする

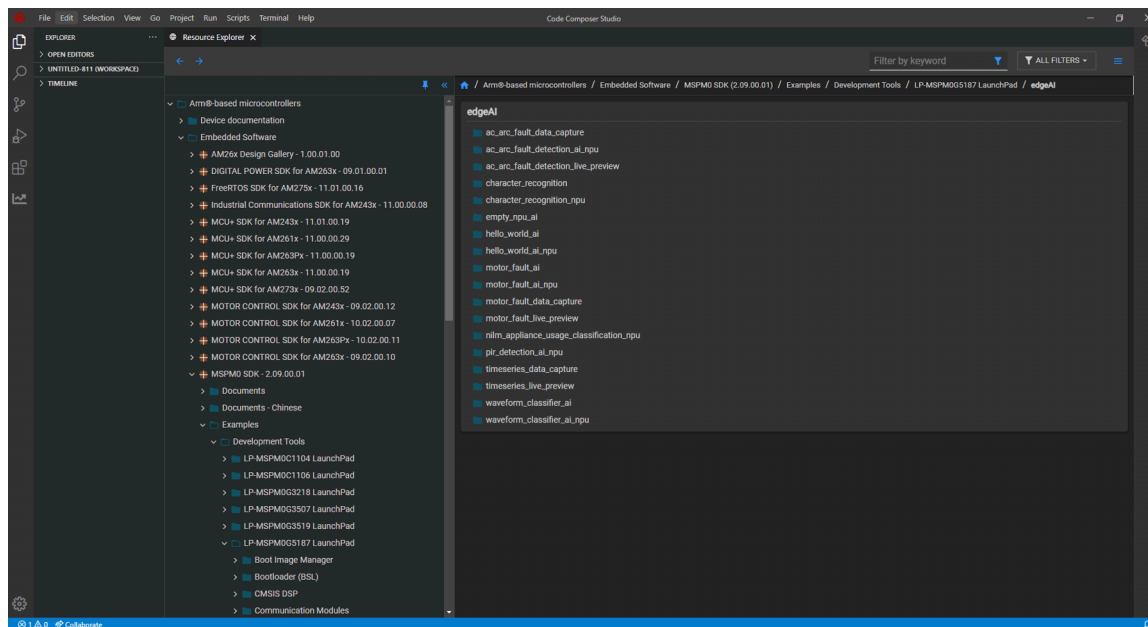


図 2-2. Code Composer Studio Resource Explorer での MSPM0 SDK のインストール

ステップ 2: ハードウェアに接続する

- TIDA-010971 ボードを LP-MSPM0G5187 LaunchPad ヘッダに接続する
- ピンの位置合わせを確認する (ピン配置についてはセクション 3.1 を参照)
- USB Type-C ケーブルを LaunchPad に接続する

警告

AC 主電源電圧を使用してテストする場合は、必ず LaunchPad とコンピュータの間に USB アイソレータを接続してください。TIDA-010971 は、危険な電圧で動作します。

LP-MSPM0G5187 LaunchPad に接続された TIDA-010971 ボード。ヘッダの適切な位置合わせと USB アイソレータの接続に注意してください。

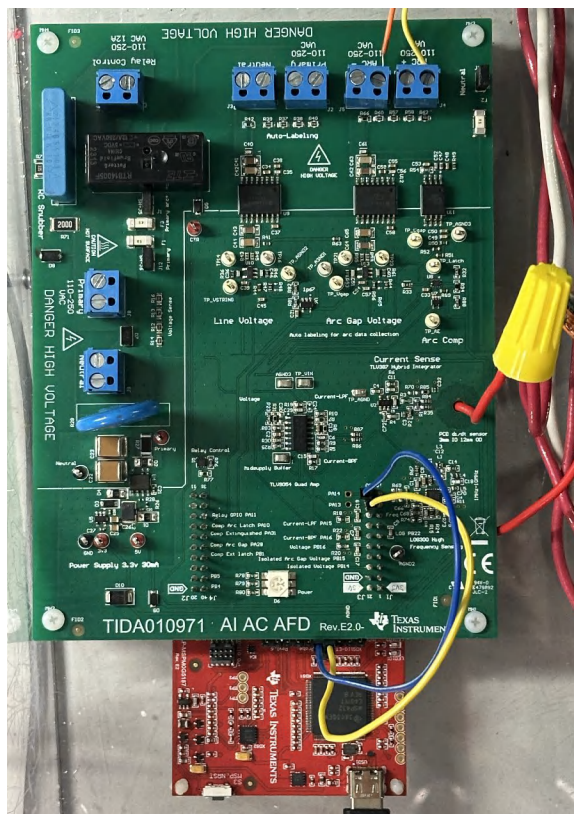


図 2-3. LP-MSPM0G5187 LaunchPad に接続された TIDA-010971 ボード

ステップ 3: プロジェクトのインポート

- CCS で、「File」→「Import」→「CCS Projects」の順に選択する
- SDK のインストールをブラウズする:<SDK_PATH>/examples/ac_arc_fault_detection_ai_npu/LP_MSPM0G5187 /
- プロジェクトを選択し、「Finish」をクリックする

ステップ 4: ビルドとフラッシュを行う

- プロジェクトを右クリック→「Build Project」(または Ctrl+B) を選択する
- ビルドがエラーなしで完了したことを確認する
- デバッグ (F11) をクリックしてフラッシュし、デバッグを開始する
- 「Resume」(F8) をクリックしてアプリケーションを実行する

ステップ 5: 動作を確認する

表 2-1. LED ステータス テーブル

LED 状態	意味
赤の ON	アーク故障検出あり
両方オフ	システムが作動中、アーク検出なし

次のステップ:

デモの実行後:

表 2-2. 次のステップ:

目標	セクション
独自のモデルをトレーニングする	[セクション 6:モデル開発] (#6-model-development)
ファームウェアを変更する	[セクション 7:ファームウェアの実装] (#7-firmware-implementation)
消費電力の最適化	[セクション 10:消費電力の最適化] (#10-power-optimization)
デバッグの問題	[セクション 12:トラブルシューティング] (#12-troubleshooting)

3 技術的な背景

アーク故障の電気的特性

アーク故障は、正常な負荷動作とは異なる独自の電気的シグネチャを生成します。

時間ドメイン特性:

- ゼロ交差での電流不連続性 (電圧降下時にアークは消滅)
- 高周波ノイズを重畳した状態の不規則な電流波形
- 電流振幅は正常な通常の負荷 (直列アーク) よりも小さい可能性がある
- ショルダー現象:アーク再点火時にゼロ交差点付近の電流ショルダー

周波数ドメイン特性:

- DC から数十 MHz までの広帯域ノイズ
- 時変ピンク ノイズ スペクトル ($1/f$ ロールオフ)
- 広い周波数範囲にわたって存在するアーク エネルギー
- 正常な通常のスイッチング過渡と比較して明確なスペクトル特性

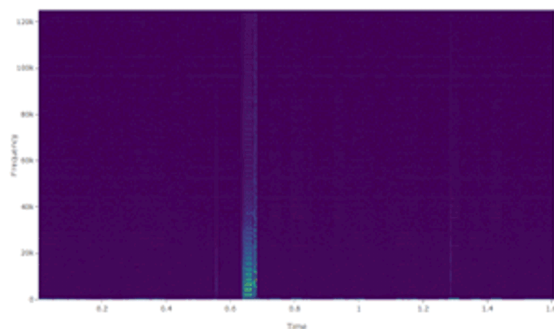


図 3-1. 正常な周波数スペクトル

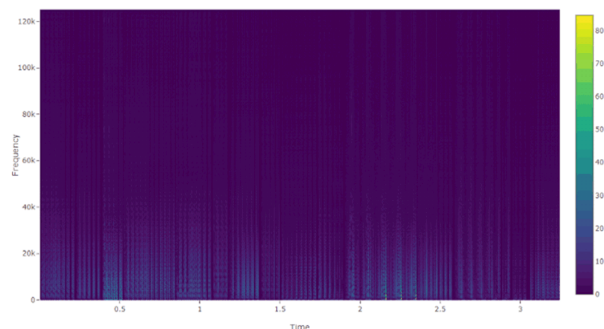


図 3-2. アーク故障の周波数スペクトル

表 3-1. アーク シグネチャと負荷ノイズとの関係

特性	アーク故障	正常な通常負荷 (モーター / 調光器)
時間パターン	ランダムで混沌としている	定期的で予測可能
ゼロ交差動作	消火 / 再点火	スムーズな遷移
エネルギー分布	スペクトル全体で均一	高調波に集中

信号処理の概要 (TIDA-010971)

TIDA-010971 は、帯域幅が 40MHz を上回る PCB ログスキー コイルを使用するマルチチャネル センシング手法を実装しています。

表 3-2. 信号チャネル

チャネル	周波数範囲	目的	ML 機能の使用法
バンドパス電流	5 ~ 50kHz	可聴範囲 / 超音波範囲のアーキ ノイズ	FFT 機能用の 1 次入力
LOG300 HF 出力	1 ~ 10MHz	高周波のアーキ エネルギー	アーキ エネルギーの大きさ
ローパス電流	DC-60Hz	基本電流振幅	負荷レベル検出
ライン電圧	60Hz	ゼロ交差リファレンス	位相同期

主な設計コンセプト

詳細な回路図、部品の値、設計の計算については、『[TIDA-010971 設計ガイド](#)』を参照してください。

- PCB ログスキー コイル: 差動コイルは di/dt 信号を出力し、40MHz 超の自己共振を使用して広帯域のアーキ検出を実現
- ハイブリッド積分器: TLV387 ベースの回路は di/dt を、68.6dB のゲインで電流比例信号に変換
- LOG300 検出器: MHz 周波数のアーキ エネルギーを検出するために 98dB のダイナミックレンジを実現

特徴抽出パイプライン

ファームウェアは、未加工 ADC サンプルから周波数ドメインの特徴を抽出します。

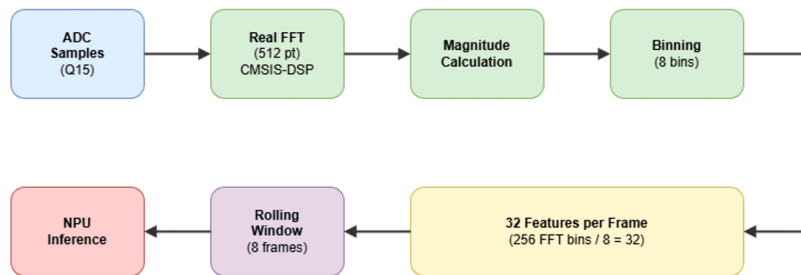


図 3-3. 特徴抽出パイプライン

タイミングの関係

- ADC サンプリング間隔: 9.3 μ s (1/107kHz、約 9.3 μ s)
- フレーム周期: 512 サンプル $\times$ 9.3 μ s は約 4.76ms に相当
- ローリング ウィンドウ: 8 フレーム $\times$ 4.76ms は、時間的コンテキストの約 38ms に相当
- 推論レート: フレームごとに 1 回 (約 4.76ms)

構成パラメータ (user_input_config.h):

注

ローリング ウィンドウに関する注記: ここでの「ウィンドウ」という用語は、FFT ウィンドウ機能ではなく、連続する 8 つのフレームの時間的スライディング ウィンドウを指します (例えば、ハニング、ハミング)。新しいフレームごとにウィンドウが順方向にシフトし、アーキ検出用の連続的な時間コンテキストを示します。

```
#define FE_FRAME_SIZE 512           // ADC samples per frame
#define FE_FEATURE_SIZE_PER_FRAME 32 // Output features per frame
#define FE_NUM_FRAME_CONCAT 8       // Frames in rolling window
#define FE_BIN_SIZE 8               // FFT bins per feature bin
#define FE_BIN_OFFSET 1             // Skip DC component
#define FE_COMPLEX_MAG_SCALE_FACTOR 5 // Magnitude scaling
```

処理ステップ:

- 実数 FFT: ARM CMSIS-DSP を使用した 512 ポイント FFT (arm_rfft_q15)
- 複素振幅: 各周波数ビンの振幅を計算
- DC 除去: DC コンポーネント成分をスキップ (FE_BIN_OFFSET = 1)
- BIN 平均 8 つの隣接する FFT ビン → 32 の特徴
- 正規化: INT8 の範囲にスケール調整 (-128 ~ 127)
- フレーム連結: ローリング ウィンドウで 8 フレームをスタック → 合計 256 個の特徴

処理ステップ

ML モデル アーキテクチャ

MSPM0 クラスのデバイスでメモリ、レイテンシ、精度に関する広範な要件に対応するため、アーク故障検出ソリューションには、軽量の畳み込みニューラル ネットワーク (CNN) モデルのファミリーが用意されています。これらのモデルは、共通の 3 層畳み込みバックボーンを共有しており、違いはチャンネルの深度のみであるため、超低フットプリントから高精度の構成までスケラブルに展開できます。ニューラル ネットワークは、抽出された特徴をアーク故障または正常として分類します。

表 3-3. モデルアーキテクチャの比較

注: 以下の性能指標は、入力寸法 N = 1、C = 1、H = 256、W = 1 で測定されたものです

モデル名	パラメータ	フラッシュ サイズ (B)	SRAM サイズ (B)	推論時間	精度	推奨事項
ArcFault_model_200_t	≒200	3652	900	197.82us	99.60%	最もシンプル、最小、最高速のモデル。
ArcFault_model_300_t	≒300	3864	1644	246.91μs	99.60%	700 や 1400 よりも高速だが、複雑さの少ないデータも処理。
ArcFault_model_700_t	≒800	4496	1644	288.36μs	99.42%	大きなモデル、推論速度と占有メモリの間のスイートスポット。
ArcFault_model_1400_t	≒1600	5564	2408	713.31μs	99.88%	最も正確なモデルであり、複雑なデータシナリオに使用 (推奨)

上記のアーキテクチャモデルは独自仕様であるため、GUI ベースの Edge AI Studio でのみ利用できます。

アーキテクチャ検出用の推奨 1D CNN アーキテクチャ (アーキテクチャモデル 1400t モデル)

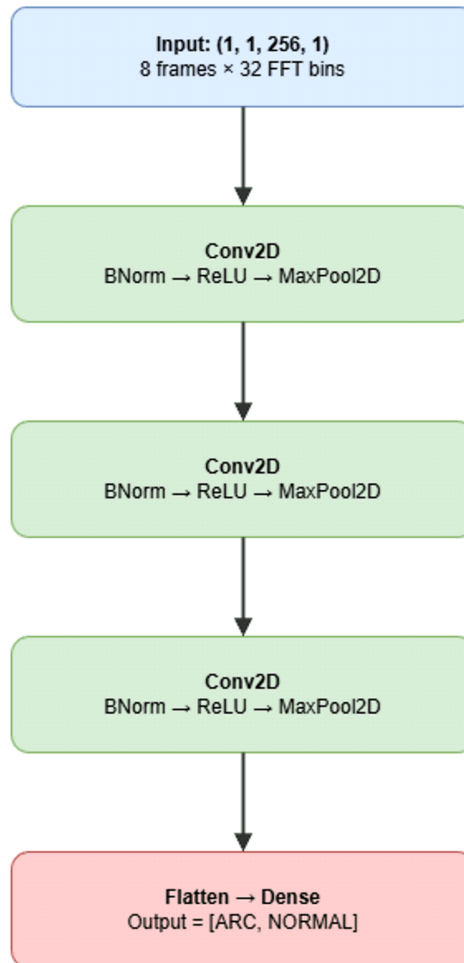


図 3-4. アーク故障検出用の推奨 1D CNN アーキテクチャ

NPU アクセラレーション

MSPM0G5187 には、ニューラル ネットワークの推論を高速化する TinyEngine™ NPU が内蔵されています。

NPU の特長:

- 一般的な NN 操作のハードウェア アクセラレーション
- モデル格納用の命令およびパラメータ メモリ
- 割り込み駆動完了信号
- TI のニューラル ネットワーク コンパイラ (TI-NNC) による TVM ランタイム統合

TI-NNC との統合

TI のニューラル ネットワーク コンパイラでの生成物:

- model.a: コンパイル済みモデル ライブラリ
- tvngen_default.hC インターフェイス ヘッダ
- user_input_config.h: 特徴抽出の設定

4 システム要件

ハードウェア要件

表 4-1. ハードウェア要件

部品	部品番号	説明
マイコン / LaunchPad	LP-MSPM0G5187	MSPM0G5187 LaunchPad 開発キット
USB ケーブル	USB Type-C	プログラミングとデバッグ用
USB アイソレータ	任意の USB 2.0 アイソレータ	AC テスト中の安全のために必須

主要コンポーネント

TIDA-010971 の仕様

表 4-2. TIDA-010971 の仕様

パラメータ	値
消費電力	7-17mA
Rogowski 帯域幅	\50Hz ~ 40MHz
Rogowski 感度	6kHz 時に最大 0.4mV/A
バンドパス フィルタ	5kHz ~ 50kHz
LOG300 HF 検出	1MHz ~ 10MHz
最大電流センシング	320A (クリッピングなし)

LaunchPad ピンの接続

表 4-3. LaunchPad ピンの接続

ピン	ADC チャンネル	信号の説明
PA15	ADC0_15	ローパス フィルタ付き電流 (60Hz 基本)
PA21	ADC0_24	LOG300 出力 (HF アーク エネルギー)
PB14	ADC0_21	絶縁ライン電圧 (AMC3330)
PA31	GPIO	ARC インジケータピン (設定:ARC あり、未設定:ARC なし)
PA11	GPIO	リレー制御出力
PB4	GPIO	赤色の LED インジケータ
PB5	GPIO	緑色の LED インジケータ

TIDA-010971 主要コンポーネント

TIDA-010971 で使用される専用アナログ コンポーネントには、TLV387 ハイブリッド積分器向けゼロドリフト オペアンプ、TLV9054 バンドパスおよびローパス フィルタ処理用クワッド オペアンプ、LOG300 高周波対数検出器 (98dB ダイナミックレンジ)、AMC3330 絶縁型アンプ、UCC28881 電源用オフライン降圧コンバータなどがあります。

IC の詳細な仕様、回路図、設計ファイルについては、『[TIDA-010971 設計ガイド](#)』を参照してください

ソフトウェア要件

表 4-4. ソフトウェア要件

ソフトウェア	バージョン	目的
MSPM0 SDK	2.10.00 以降	デバイスドライバとライブラリ
TI Edge AI Studio	1.6.0 以降	モデルのトレーニングと導入用の GUI ツール
モデル メーカー	R1.3 以降	モデルのトレーニングと導入用の CLI ツール
ARM CMSIS-DSP	SDK の一部	FFT と DSP の機能
Python	3.10 以降	モデルトレーニング / データ ラベリング スクリプト

モデル要件

表 4-5. モデル要件

パラメータ	仕様
入力寸法	8 フレーム × 32 FFT ビン = 合計 256 の特徴
出力形状	(1, 2) - [arc_confidence, normal_confidence]
量子化	INT8 (必須 — Cortex-M0+ に FPU なし)
モデル サイズ	約 14KB のパラメータ
テンソル アリーナ	約 8KB の SRAM

テンソル形状の説明

- バッチ (1): 一度に 1 つの推論
- チャンネル (1): 単一入力チャンネル (グレイスケールタイプの表現)
- 高さ (1024): 連結特徴ベクトル (8 フレーム × 32 の特徴)
- 幅 (1): 単一要素の幅 (2D としてのベクトル表現)

5 ハードウェア設計

システム概要

完全なアーク故障検出システムは、LP-MSPM0G5187 LaunchPad に接続された TIDA-010971 アナログ フロント エンドで構成されています。

エッジ AI アーク故障検出用に LP-MSPM0G5187 に接続された TIDA-010971 アナログ フロント エンドを示す包括的なシステム ブロック図

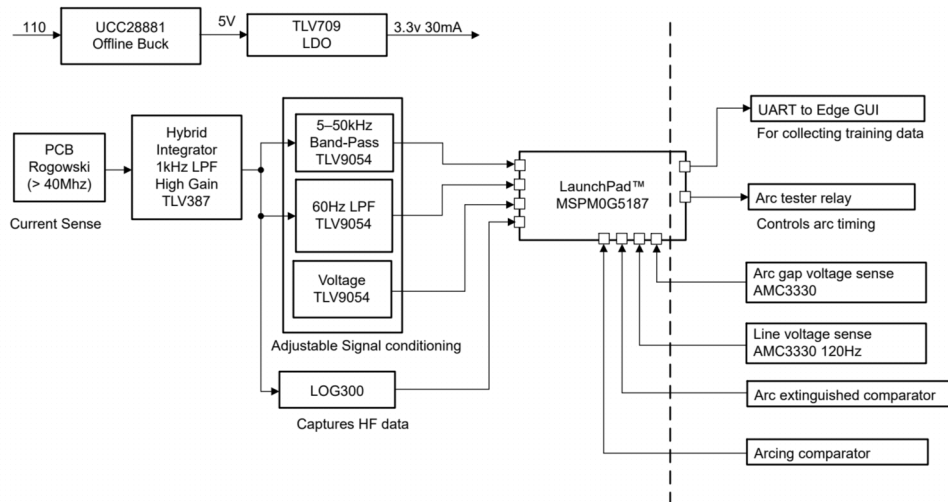


図 5-1. TIDA-010971 システム ブロック図

主要部品

- TIDA-010971: アナログ シグナル コンディショニング、電源、絶縁機能を装備
- LP-MSPM0G5187: ADC サンプリング、特徴抽出、NPU 推論、トリップ決定を実行

詳細なシステム回路図、PCB レイアウト、設計ファイルについては、『TIDA-010971 設計ガイド』を参照してください。

ハードウェア設定

ボードの接続

BoosterPack ヘッドを使用して TIDA-010971 を LP-MSPM0G5187 に接続します。電源投入前にピン配置を確認します。さまざまなアークまたは正常のシナリオを生成するため、アーク ジェネレータのセットアップをアナログ フロント エンドに接続します。

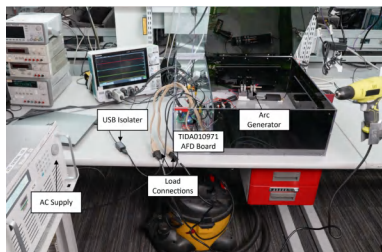


図 5-2. アーク ジェネレータ試験セットアップ

安全に関する検討事項

警告

危険な電圧

この設計は、重傷や死亡の原因となる可能性のある AC 主商用電源電圧 (110 ~ 250VAC) で動作します。すべての安全に関する注意事項に従ってください。

1. 必要な USB 絶縁: AC テスト中は、LaunchPad と PC の間に必ず USB アイソレータを接続してください
2. 筐体: 必ず、適切に接地され、絶縁された筐体の中で動作させてください
3. 電流制限: 開発中は、電流制限された AC 電源を使用してください
4. ノータッチ ルール: 通電中は、ボードや接続部に絶対に触れないでください
5. 放電時間: 取り扱い前にコンデンサを放電させてください (5 分超)
6. 資格のある担当者: AC テストは、必ず資格のある電気技術者が実施してください

6 モデル開発

このセクションでは、TI のエッジ AI ツールを使用してアーク故障検出 ML モデルを開発して導入する方法について説明します。

開発ワークフローの概要

データ収集から導入までのエンドツーエンドのモデル開発ワークフロー



図 6-1. モデル開発ワークフロー

データ収集

ハードウェア設定

- データ キャプチャ ファームウェアをフラッシュ: `examples/ac_arc_fault_data_capture/LP_MSPM0G5187/`
- BoosterPack ヘッダ経由で TIDA-010971 を LP-MSPM0G5187 に接続
- USB を接続 (AC テスト用のアイソレータ経由)
- シリアル接続: 5,820,000 ボー、バイナリ フレーム

データ収集シナリオ

管理された条件下で各クラスのデータを収集します。各負荷タイプには、最低 20000 回以上の半周期キャプチャが含まれています (正常構成とアーク構成で約 3 分間を均等に分割)。

表 6-1. クラス 1: 正常動作 (誤切断なし)

負荷タイプ	説明	キャプチャ
誘導性	AC リレー、コンプレッサ モーター、掃除機真、シヨップ ツール	10000 またはそれ以上
LED	ハロゲン照明、600W ~ 1000W、調光器付き	10000 またはそれ以上
SMPS	600W AC/DC	10000 またはそれ以上
混合あり	上記の組み合わせで、5A の抵抗性負荷を使用	10000 またはそれ以上

表 6-2. クラス 2: アーク故障マスキング負荷

アーク タイプ	説明	キャプチャ
設定 A	アーク ジェネレータを使用して、負荷を直列で接続	10000 またはそれ以上
設定 B	アーク ジェネレータを使用して、負荷を 5A の抵抗性負荷に直列および並列で接続	10000 またはそれ以上
設定 C	アーク ジェネレータを使用して、負荷を 5A の抵抗性負荷に並列で接続	10000 またはそれ以上
設定 D	負荷と 5A の抵抗性負荷の両方でアーク ジェネレータを使用	10000 またはそれ以上

データのラベリング

キャプチャされたデータはラベルにより 2 つのクラスに分類されます。ラベリング ツールには、未加工の波形 CSV から **ARC** および **NORMAL** のトレーニング データを生成するための構成可能なモードとして、フレーム モードとウィンドウ モードが用意されています。どちらのモードも、アーク電圧ベースのヒューリスティックを使用してグラウンドトゥールス ラベルを判定しますが、機械学習モデルには依存しません。どちらのモードも異なるデータセットとフロント エンド アーキテクチャへの対応にサポートされていますが、フレーム モードはイベントの継続性をより適切に維持し、よりクリーンなクラス境界を生成することができるため、ほとんどのアプリケーションでデフォルトとして推奨されています。

フレーム モード (推奨)

目的: フレーム モードは、自然なアーク イベント期間をキャプチャし、実際の電氣的挙動を忠実に反映する可変長の **ARC** および **NORMAL** のセグメントを生成するように設計されています。時間連続性の保持が重要な **LOG300** ベースのデータセットやシナリオに特に適しています。

動作の仕組み:

1. 波形が固定長のフレームに分割されます
2. 各フレームについて、アーク ギャップ電圧の統計情報 (ピークと **RMS** など) が、構成可能なスレッシュホールドに照らして評価されます
3. 連続するフレーム全体に時間平滑化を適用することにより、絶縁スパイクとスイッチング過渡を抑制します。ステートマシン ベースのマージにより、連続する **ARC** または **NORMAL** のフレームをより長いセグメントに結合します。ノイズの多い短いフラグメントでのトレーニングを避けるために、最小期間を下回るセグメントは破棄されます。確認された **ARC** 領域が重複しているフレームは、正常サンプルを確認するために **NORMAL** 出力から除外されます

出力特性

1. 可変長の **ARC** および **NORMAL** のセグメント
2. 強力なノイズ抑制とラベル ジッタの低減
3. アーク イベントの自然な開始 / 停止境界を保持
4. 構成可能な保存チャネル (**LOG300** エンベロープや電流など)、ラベリング ロジックは常に電圧を使用

使用に適した状況:

1. **LOG300** またはエンベロープベースのフロント エンド
2. ラベルの忠実度とイベントの継続性が求められる用途
3. トランジェント ノイズまたは境界アーチファクトの影響を受けやすいモデルのデータセット作成

ウィンドウ モード (推奨)

目的:ウィンドウ モードでは、スライディング ウィンドウ アプローチを使用して、固定長のトレーニング サンプルを生成します。これは、均一な入力サイズが必要な場合や、一定長の機能が想定されるアーキテクチャのデータセットを構築する場合に便利です。

動作の仕組み:

1. 信号は最初にフレームに分割されます。
2. 連続する **N** 個のフレームのウィンドウが、波形を横切ってスライドします。
3. 各ウィンドウ内で、(電圧スレッシュホールドに基づいて) アーク状フレームの数がカウントされます。
4. カウントが構成可能なスレッシュホールドを超えると、ウィンドウ全体に「**ARC**」というラベルが付けられます。電圧レベルが十分に低く安定している場合は、「**NORMAL**」というラベルが付けられます。それ以外の場合、ウィンドウは破棄されます。
5. ラベル付けされたウィンドウに対応する抽出された波形は、固定サイズのサンプルとして保存されます。

出力特性:

1. 固定長の **ARC** および **NORMAL** ウィンドウ
2. バッチ処理が容易で、モデルの入力寸法に一貫性がある
3. 信頼性を維持するために、あいまいなウィンドウを破棄することがある
4. ラベル付けは電圧駆動型を維持し、保存されたチャネルは通常、電流

使用に適した状況

1. 均一なサンプル長を必要とするパイプライン。
2. イベント継続性の要件よりもバッチ処理の完全なシンプルさが重視されるシナリオ。

推奨事項

どちらのモードも完全にサポートされており、同じ電圧ベースのグラントゥールース ロジックを共有しています。ただし、フレーム モードがデフォルトで推奨されます。その理由は、実世界のアーク イベントの継続性を確実に保持し、時間平滑化によってノイズに起因する誤ったラベル付与を低減し、アーク隣接領域を明示的に除外することでよりクリーンな **NORMAL** サンプルを生成することができるためです。ウィンドウ モードは、固定サイズの入力または迅速なデータセットの標準化が主な要件である場合に有用です。

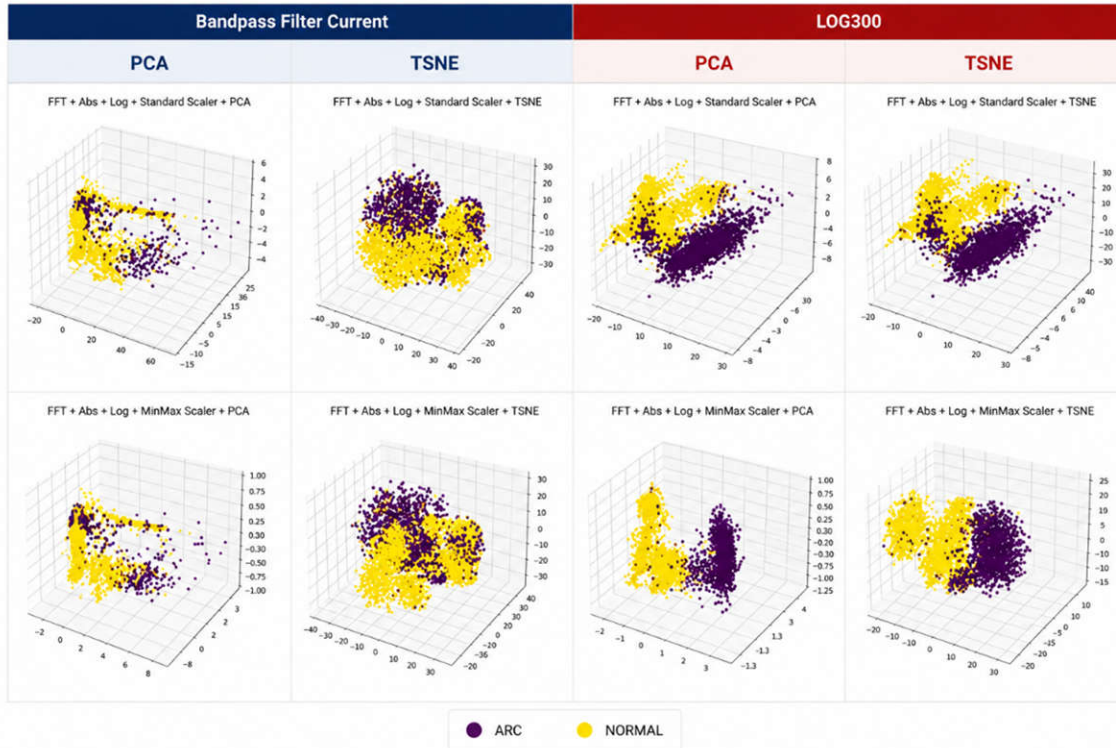
詳細については、`Tinyml-modelzoo -> examples -> ac_arc_fault -> readme.md` を参照してください

データセットの分析と選択

初期段階の評価として **Goodness of Fit (GOF)** プロットを使用し、モデルのトレーニングの前にデータセットの特徴分離性を評価しました。これらのプロットは、FFT、ウェーブレット変換、スケーリング、PCA、t-SNE など、さまざまな前処理の組み合わせにおける **ARC** および **NORMAL** のサンプルの分布を比較するのに役立ちました。GOF プロットの強力なクラス分離は、選択したシグナル チェーンと特徴抽出法が分類に適していることを示していますが、クラス間の重なりはさらなる調整が必要であることを示唆しています。

バンドパスでフィルタ処理された電流のデータセットでは、GOF プロットは、特に FFT ベースの前処理で **ARC/NORMAL** の重なりを顕著に示し、アーク関連の一部の機能がマスクされていることを示しました。それに対して、LOG300 ベースのデータセットはより明確なクラス クラスタを生成し、特徴分離性が向上し、ダウンストリーム モデルのトレ

ーニングに適していることを示唆しました。データセットの GOF、特徴抽出の品質、GOF グラフ、ファイル レベルの分類サマリーの詳細については、『TI Tiny ML Tensorlab User Guide』の「[Advanced Features](#)」に記載されています。
バンドパスでフィルタ処理された電流と LOG300 のデータセットの GOF 比較



モデル アーキテクチャ

セクション 4.4 に示したアーク故障モデルは、AC アーク故障検出のモデルとして推奨されるものですが、分類には、さまざまな汎用時系列モデルを使用することもできます。

汎用時系列モデル テンプレートは、TI Edge AI Studio (GUI) と TensorLab (CLI) の両方で利用でき、組み込み ML アプリケーションをすばやく開発、評価することができます。これらのテンプレートは、カスタム データセットにより最小限の構成作業でトレーニングできる、NPU 互換のすぐに使用可能なアーキテクチャを備えています。

そのようなリファレンス アーキテクチャの 1 つに CLS_6K_NPU モデルがあり、これは深さによる分離が可能な軽量の畳み込みニューラル ネットワークです。このモデルと他の幅広いリファレンス分類アーキテクチャは、TI Edge AI Studio (GUI) 内で利用でき、また、CLI ツールで提供される Model Zoo から利用できます。

```
import torch
import torch.nn as nn
import torch.nn.functional as F

class CLS_6K_NPU(nn.Module):
    def __init__(self, input_features=512, variables=1, num_classes=2):
        super(CLS_6K_NPU, self).__init__()

        self.bn0 = nn.BatchNorm2d(variables)

        self.conv1 = nn.Conv2d(variables, 16, kernel_size=(3, 1), stride=(4, 1), padding=0, bias=False)
        self.bn1 = nn.BatchNorm2d(16)

        self.pool1 = nn.MaxPool2d(kernel_size=(3, 1), stride=(2, 1))
        self.dw3 = nn.Conv2d(16, 16, kernel_size=(3, 1), stride=(1, 1), padding=0, groups=16, bias=False)
        self.bn3a = nn.BatchNorm2d(16)

        self.pw3 = nn.Conv2d(16, 32, kernel_size=(1, 1), stride=(1, 1), padding=0, bias=False)
```

```
self.bn3b = nn.BatchNorm2d(32)

self.pool2 = nn.MaxPool2d(kernel_size=(3, 1), stride=(2, 1))

self.dw5 = nn.Conv2d(32, 32, kernel_size=(3, 1), stride=(1, 1), padding=0,
groups=32, bias=False)
self.bn5a = nn.BatchNorm2d(32)

self.pw5 = nn.Conv2d(32, 48, kernel_size=(1, 1), stride=(1, 1), padding=0, bias=False)
self.bn5b = nn.BatchNorm2d(48)

self.conv6 = nn.Conv2d(48, 16, kernel_size=(5, 1), stride=(1, 1), padding=0, bias=False)
self.bn6 = nn.BatchNorm2d(16)

self.gap = nn.AdaptiveAvgPool2d(output_size=(4, 1))
self.fc = nn.Linear(64, num_classes)

def forward(self, x):
    x = self.bn0(x)
    x = self.conv1(x)
    x = F.relu(self.bn1(x))
    x = self.pool1(x)
    x = self.dw3(x)
    x = F.relu(self.bn3a(x))
    x = self.pw3(x)
    x = F.relu(self.bn3b(x))

    x = self.pool2(x)
    x = self.dw5(x)
    x = F.relu(self.bn5a(x))
    x = self.pw5(x)
    x = F.relu(self.bn5b(x))
    x = self.conv6(x)
    x = F.relu(self.bn6(x))
    x = self.gap(x) # (B, 16, 4, 1)
    x = x.view(x.size(0), -1) # (B, 64)
    x = self.fc(x) # (B, num_classes)
    return x
```

注

この例では、PyTorch フレームワークに対応する TI Edge AI Studio GUI/CLI ツールを使用しています。

モデルのトレーニング

表 6-3. ArcFault_model_1400_t の推奨ハイパーパラメータ

パラメータ	値
バッチ サイズ	50
学習率	0.04
オプティマイザ	SGD
重量減衰	1e-5
データセットの分割	トレーニング 60%、検証 30%、テスト 10%

トレーニング オプション

- GUI: ガイド付きトレーニング ワークフローには **CCStudio™ Edge AI Studio GUI** を使用します
- CLI: コマンド ラインのトレーニングには **tinymml-tensorlab** を使用します

最高の INT8 の精度を得るために量子化対応トレーニング (QAT) を有効にします。Tinymml-tensorlab と Edge AI Studio には、8 ビット NPU 対応量子化のサポートが内蔵されています。

量子化

INT8 量子化を使用する理由

MSPM0G5187 Cortex-M0+ には、以下のハードウェア浮動小数点演算ユニットがありません。

- FP32 操作にはソフトウェア エミュレーションが必要 (非常に低速)
- INT8 操作はシングル サイクルで実行される
- NPU は INT8 操作のみに対応
- モデル サイズを最大約 4 倍縮小

量子化対応トレーニング (QAT)

QAT では、最高の精度を維持するために (通常は 1% 未満の低下)、トレーニング中に量子化をシミュレーションします。

```
import torch.quantization as quant

# Prepare model for QAT
model.qconfig = quant.get_default_qat_qconfig('fbgemm')
quant.prepare_qat(model, inplace=True)

# Train with simulated quantization
for epoch in range(num_epochs):
    train_one_epoch(model, train_loader)

# Convert to fully quantized model
model_int8 = quant.convert(model, inplace=False)
```

入力正規化

正規化パラメータはモデル固有であり、TI-NNC によって生成されます。

```
// From user_input_config.h - [MODEL_SPECIFIC]
#define FE_NORM_BIAS [MODEL_SPECIFIC] // Generated by TI-NNC
#define FE_NORM_SCALE [MODEL_SPECIFIC] // Generated by TI-NNC
#define FE_NORM_SHIFT [MODEL_SPECIFIC] // Generated by TI-NNC
```

重要: 必ず、コンパイル時に生成されたモデルの `user_input_config.h` ファイルの値を使用してください。

TI-NNC を使用したモデルのコンパイル

TI のニューラル ネットワーク コンパイラ (TI-NNC) は、トレーニングされた「onnx」INT8 モデルを、MSPM0 NPU 実行用に変換します。

コンパイルでの生成物:

- `model.a` - コンパイル済みモデル ライブラリ (14KB 以下)
- `tvmgen_default.h` - C インターフェイス ヘッダ
- `user_input_config.h` - 特徴抽出設定

CCS プロジェクトへの統合:

- 生成されたファイルを `project/model/` ディレクトリにコピー
- `model.a` をリンカ入力ファイルに追加
- ヘッダをソースに組み込む: `#include model/tvmgen_default.h`

インストール、CLI 構文、および詳細なコンパイル手順については、『TI-NNC ユーザー ガイド』を参照してください。

7 ファームウェアの実装

このセクションでは、LP-MSPM0G5187 でリアルタイムでアーク故障検出のためのファームウェア アーキテクチャについて詳しく説明します。

システム アーキテクチャ

メイン ループ ステート マシン:

メイン アプリケーション ループと割り込みハンドラを示したファームウェア ステート マシンの図

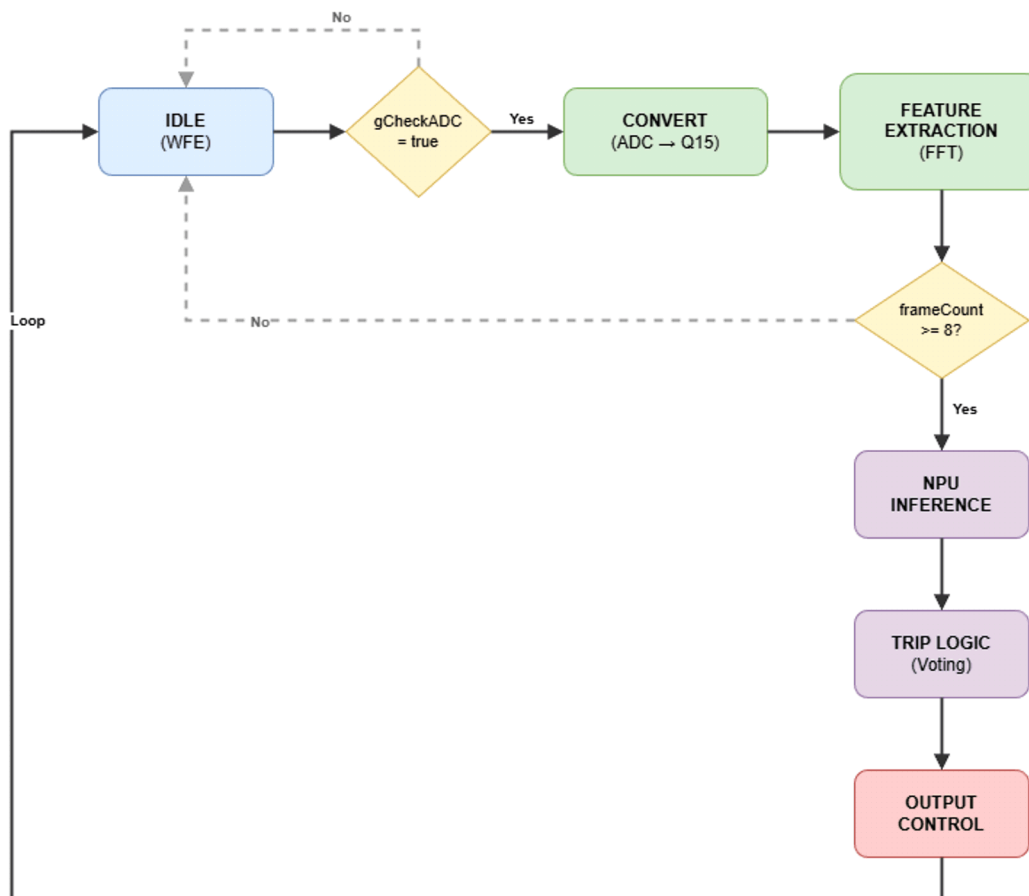


図 7-1. ファームウェア ステート マシンの図

ペリフェラルの構成

TI SysConfig を使用して MSPM0G5187 ペリフェラルを構成します。主な設定:

表 7-1. ペリフェラルの構成

ペリフェラル	構成
ADC12_0	12 ビット、SYSOSC クロック、DMA イネーブル、チャンネル ADC0_14
DMA	ピンポン転送、バッファあたり 512 サンプル
GPIO	PA11 (リレー)、PB4/PB5 (LED)、PA31 (コンパレータ入力)

メモリ マップ

表 7-2. メモリ マップ

フラッシュ メモリ (128 KB)	SRAM (32KB)
利用可能 (約 94KB)	利用可能 (約 23KB)
モデル (約 5.5 KB)	スタック (約 2 KB)
モデル (約 5.5 KB)	テンソル アリーナ (2KB)
アプリ コード (約 20KB)	特徴バッファ (1KB)
	ADC バッファ (1KB)
	FFT スクラッチ (2KB)

ADC データ アクイジション

ピンポン バッファ戦略

```
/* ADC Ping-Pong Buffers */
int16_t gADCSamplesPing[ADC_SAMPLE_SIZE]; // 512 samples
int16_t gADCSamplesPong[ADC_SAMPLE_SIZE]; // 512 samples

volatile bool gPing = true; // Current buffer indicator
volatile bool gCheckADC = false; // New data ready flag
```

CPU が 1 つのバッファを処理しているときに、DMA ではもう 1 つのバッファを使用できるため、連続サンプリングが可能です。

ログ 300 データ スケーリング

```
/* Scales ADC samples and converts them to Q15 format */
void adc_q15_scale(int16_t *samples)
{
    int16_t shift = 2048; // Mid-scale for 12-bit ADC
    for (int i = 0; i < ADC_SAMPLE_SIZE; ++i)
    {
        // Convert from unsigned 12-bit to signed Q15
        // Shift of 4 provides 16x gain for better SNR
        samples[i] = samples[i]*8
    }
}
```

特徴抽出

```
#include "feature_extract.h"

// Initialize feature extraction (call once at startup)
FE_init();

// Process each frame of ADC samples
FE_process(newSamples, gNewFeatures, 0); // varIndex=0 for single channel
```

NPU 推論

```
/* NPU Initialization */
DL_NPU_reset(NPU);
DL_NPU_enablePower(NPU);
while (!(DL_SYSCTL_getStatus() & DL_SYSCTL_STATUS_NPU_READY));

/* Input/Output tensor structures */
struct tvmgem_default_inputs tvm_if_map = { (void*) &gIF_Map[0] };
struct tvmgem_default_outputs tvm_of_map = { (void*) &gOF_Map[0] };

/* Run inference */
tvmgem_default_run(&tvm_if_map, &tvm_of_map);
```

```
/* Wait for completion */
while (!tvmgen_default_finished) { __WFE(); }

/* Output: goF_Map[0][0] = arc confidence, goF_Map[0][1] = normal confidence */
```

トリップ ロジック (投票アルゴリズム)

```
#define INFERENCE_FRAME_THRESHOLD_FOR_ARC 5

/* Circular buffer for predictions */
volatile uint8_t gPredVector[8] = {0};
volatile uint8_t gPredPtr = 0;
volatile uint16_t gPredSum = 0;

int trip_logic(int inf)
{
    /* Update circular buffer */
    gPredSum -= gPredVector[gPredPtr];

    if (inf >= 0) {
        gPredVector[gPredPtr] = 1;
        gPredSum += 1;
    } else {
        gPredVector[gPredPtr] = 0;
    }

    gPredPtr = (gPredPtr + 1) % 8;

    /* Majority voting: 5 out of 8 frames must detect arc */
    return (gPredSum > INFERENCE_FRAME_THRESHOLD_FOR_ARC) ? 1 : 0;
}
```

8 サンプルコード

利用可能な例

表 8-1.

例	ロケーション	説明
アーク故障検出 (NPU)	`examples/ac_arc_fault_detection_ai_npu/'	推論アプリケーションの完了
データ キャプチャ	`examples/ac_arc_fault_data_capture/'	トレーニング データの収集
ライブ プレビュー	`examples/ ac_arc_fault_detection_live_preview/'	リアルタイム視覚化

プロジェクトファイル構造

```
ac_arc_fault_detection_ai_npu/LP_MSPM0G5187/
├── ac_arc_fault_detection_ai_npu.c      # Main application
├── ac_arc_fault_detection_ai_npu.syscfg # SysConfig settings
├── ticlang/mspm0g5187.cmd              # Linker script
├── model/
│   ├── model.a                        # Compiled NN model
│   ├── tvmgen_default.h              # Model interface
│   └── user_input_config.h           # Feature config
```

図 8-1. プロジェクトファイル構造

9 性能ベンチマーク

注

以下に示す性能ベンチマークは、入力構成 [1,1,256,1] で ArcFault_model_1400_t を対象としたものです。

表 9-1. 性能ベンチマーク

メトリック	値	注
モデル サイズ - ArcFault (フラッシュ)	約 5564B	コンパイル済みの model.a
アプリケーション コード	約 13 KB	ライブラリを含む
テンソル アリーナ (SRAM)	約 2 KB	ランタイム ワークスペース
使用されている SRAM の合計	約 9 KB	32KB のうちの使用可能量
タイミング		
ADC サンプリング	4.75ms	107kSps で 512 サンプル
特徴抽出	約 5ms	FFT + ビニング
NPU 推論	約 0.7ms	(モデルによって異なる)
合計フレーム時間	約 7.19ms	推論サイクルあたり
エンドツーエンドのレイテンシ	<150ms	8 フレーム投票を含む
電源		
アクティブ時の消費電力 (マイコン)	約 8mW	@ 80MHz, 3.3V
システム全体	約 55mW	TIDA-010971 AFE を含む
精度		
検出精度	>99%	標準的なトレーニング済みモデル
誤検出率	0.01%	誤ってフラグが付けられた負の割合
精度	99.97%	予測された正の正しい割合
リコール	99.72%	実際の正が検出された割合
F1 - スコア	99.84%	精度とリコールのバランス。

10 消費電力の最適化

このリファレンス デザインは継続的な監視により、最小のレイテンシを実現します。これは次のようなセーフティ クリティカルな AFCI アプリケーションに最適です。

- ADC は 107kSps で連続的に動作
- NPU 推論は、約 9.3ms ごとに実行
- 常時オン動作

バッテリー駆動型のアプリケーションの場合、デューティ サイクル動作を採用すると消費電力を約 50% 低減できますが、それに伴って検出レイテンシが長くなります。これは UL 1699 準拠のためには推奨されません。

11 テストと検証

必要な機器

表 11-1. 必要な機器

機器	目的
オシロスコープ	波形キャプチャ (≥100MHz)
電流プローブ	電流測定 (DC-50MHz)
USB アイソレータ	PC 保護 (2500V 絶縁)

安全上の警告

警告

アーク生成の安全性:アーク生成には、感電、重度の火傷、火災、有毒ガスなどの極度の危険が伴います。

アーク テストは、資格のある電気エンジニアが適切な PPE (個人用保護具) を着用して行う必要があります。安全観察者がいる状態で、防火環境で作業します。電流制限付き AC 電源を使用してください。

基本的なテスト手順

- 電源オン:3.3V レールを確認し、マイコンが起動することを確認する (LED 表示)。注:LaunchPad を使用する場合は、ボードに他の入力から電力を供給しないでください。
- 通常動作:さまざまな負荷を印加し、誤切断がないことを確認する (各 5 分)
- アーク検出:制御されたアークを生成し、100ms 以内に検出を確認する
- マスキング負荷:調光器、掃除機、ドリルをテストし、誤切断がないことを確認する

12 トラブルシューティング

ビルド エラー

表 12-1. ビルド エラー

エラー	設計
`-lmodel` が見つかりません`	model/ ディレクトリ内の model.a を検証し、リンカ パスを確認してください

ランタイムの問題

表 12-2. ランタイムの問題

症状	設計
入力に関係なく同じ出力	正規化パラメータがトレーニングと一致していることを検証し、Q15 変換を確認してください

検出の問題

表 12-3. 検出の問題

問題	設計
アーク検出の欠落	低い投票スレッショルド (4/8)。センサの帯域幅を確認してください

注

NPU が内蔵されているのは MSPM0G5187 のみです。他の MSPM0 バリエーションでは、NPU アクセラレーション推論は使用できません。

13 まとめ

このアプリケーション ノートでは、エッジ AI を使用してニューラル プロセッシング ユニット (NPU) を内蔵した低コストの MSPM0G5187 マイコンに AC アーク故障検出を実装するための包括的なリファレンス デザインを紹介します。このデザインでは、安全性に関する重要な課題に対処します。アーク故障は年間 30,000 件を超える家庭火災を引き起こしていますが、従来のサーキット ブレーカやヒューズではそれらを検出できません。直列アークは正常な負荷よりも流れる電流が少なく、過熱保護デバイスが監視しない周波数 (kHz ~ MHz) で動作する可能性があるためです。

この提案の設計では、機械学習推論を活用し、固定アナログ回路の制限を克服します。ハードウェア アクセラレーション形式のニューラル ネットワーク処理と、TIDA-010971 アナログ フロント エンドによる高度なシグナル コンディショニングを組み合わせることで、このシステムは UL 1699 安全規格に準拠した、99% を超える検出精度、10ms 未満の応答時間、99% を超えるマスキング負荷の特異性を実現しています。このアーキテクチャは 4 ファミリの CNN モデル セットを採用し、推奨される ArcFault_model_1400_t は 99.88% の精度を実現しながら、消費電力はフラッシュ メモリで 5.6KB、SRAM で 2.4KB のみです。

ファームウェアの実装では、107kSps の連続 ADC サンプルングと、ピンポン DMA 転送により、中断のないデータ アクイジションを実現します。512 ポイント FFT を使用したリアルタイム特徴抽出では、推論サイクルごとに 256 の時間と周波数の特徴を生成し、これは NPU によって約 0.7ms で処理されます。多数決アルゴリズム (8 フレームのうち 5 フレーム) は、堅牢な検出を保証すると同時に、誤切断を最小限に抑えます。エンドツーエンドの合計レイテンシは 150ms 未満にとどまり、マイコンでのアクティブ消費電力は約 8mW、システム全体では 55mW です。このリファレンス デザインには、データ収集やラベリングの戦略から、TI のニューラル ネットワーク コンパイラを使用した量子化、トレーニング、導入に至るまで、モデル開発に関する包括的なガイダンスが組み込まれています。フレーム モードとウィンドウ モードの両方のラベリング アプローチがサポートされており、異なるデータセットとフロント エンド アーキテクチャに対応できます。提供されているサンプル、ファームウェア コード、性能ベンチマークを使用すると、米国電気工事規格 (NEC) 要件と UL 1699 規格に準拠しながら、最小限の開発作業で住宅用および商用の電気システム向けの量産 AFCI 製品を開発することができます。このリファレンス デザインは、AFCI 開発のための強力な基盤となりますが、量産デバイスでは UL 認定の正式なテストと検証が必要です。モジュール型アーキテクチャとスケーラブルなモデル オプションにより、超低フットプリントの実装から複雑な電気環境向けの高精度構成まで、幅広い性能要件とコスト要件に対応した導入が可能です。

14 参考資料

ハードウェア

1. テキサス・インスツルメンツ、[回路ブレーカ向けエッジ AI を用いた AC アーク故障検出リファレンス デザイン](#)、デザイン ガイド
2. テキサス インスツルメンツ、[MSPM0G5187 評価基板 EVM ユーザー ガイド](#)。
3. テキサス インスツルメンツ、『[MSPM0G5187、TinyEngine™ NPU 搭載、ミクストシグナル マイクロコントローラ](#)』、データシート

ソフトウェア

1. テキサス インスツルメンツ、[MSPM0-SDK](#)、ソフトウェア
2. テキサス インスツルメンツ、[Edge AI Studio](#)、ソフトウェア。
3. テキサス インスツルメンツ、『[TI Neural Network Compiler for MCUs User's Guide -2.1.1 LTS](#)』、Web ページ
4. テキサス インスツルメンツ、[Texas Instruments/tinymml-tensorlab](#)、Web ページ。

規格

1. UL 1699、アーク故障回路遮断器、規格。
2. NEC 第 210.12 条 AFCI 保護要求事項、規格。
3. IEC 62606、AFDD の一般要求事項、規格。

15 用語

表 15-1. 用語

用語	定義
AFCI	アーク故障回路遮断器
DMA	ダイレクト メモリ アクセス
エッジ AI	エッジ デバイスでの機械学習による推論
FFT	高速フーリエ変換
NPU	ニューラル プロセッシング ユニット
Q15	15 の小数ビットを持つ固定小数点形式
QAT	量子化対応トレーニング
TI-NNC	TI ニューラル ネットワーク コンパイラ

16 よくある質問

Q: この設計を認証済み AFCI 製品に使用できますか？

A: これはリファレンス デザインです。量産 AFCI では、UL 1699 の正式な認証テストが必要です。

Q: この機能は他の MSPM0 デバイスで動作しますか？

A: NPU アクセラレーション推論には MSPM0G5187 (NPU を内蔵した MSPM0 のみ) が必要です。他のバリエーションは、ハードウェア アクセラレーションなしでソフトウェアのみの推論を実行することができます。

Q: 必要なトレーニング データはどれくらいですか？

A: クラスあたりごとに 100 キャプチャ以上です。生産モデルでは、通常、クラスあたり 500 以上のキャプチャを使用します。

Q: 別の電流センサを使用できますか？

A: はい。TMCS1123 ホール エフェクト センサまたは電流トランスは、適切なシグナル コンディショニングと連携して動作します。アーク検出用の DC-MHz 帯域幅を確保してください。

Q: 推論結果が常に同じになるのはなぜですか？

A: user_input_config.h の正規化パラメータがトレーニングと一致していることを確認してください。Q15 変換とその model.a がアーキテクチャと一致していることを検証してください。

重要なお知らせと免責事項

TI は、技術データと信頼性データ (データシートを含みます)、設計リソース (リファレンス デザインを含みます)、アプリケーションや設計に関する各種アドバイス、Web ツール、安全性情報、その他のリソースを、欠陥が存在する可能性のある「現状のまま」提供しており、商品性および特定目的に対する適合性の黙示保証、第三者の知的財産権の非侵害保証を含むいかなる保証も、明示的または黙示的にかかわらず拒否します。

これらのリソースは、TI 製品を使用する設計の経験を積んだ開発者への提供を意図したものです。(1) お客様のアプリケーションに適した TI 製品の選定、(2) お客様のアプリケーションの設計、検証、試験、(3) お客様のアプリケーションに該当する各種規格や、その他のあらゆる安全性、セキュリティ、規制、または他の要件への確実な適合に関する責任を、お客様のみが単独で負うものとし、TI は一切の責任を拒否します。

上記の各種リソースは、予告なく変更される可能性があります。これらのリソースは、リソースで説明されている TI 製品を使用するアプリケーションの開発の目的でのみ、TI はその使用をお客様に許諾します。これらのリソースに関して、他の目的で複製することや掲載することは禁止されています。TI や第三者の知的財産権のライセンスが付与されている訳ではありません。お客様は、これらのリソースを自身で使用した結果発生するあらゆる申し立て、損害、費用、損失、責任について、TI およびその代理人を完全に補償するものとし、TI は一切の責任を拒否します。

TI の製品は、[TI の販売条件](#)、[TI の総合的な品質ガイドライン](#)、[ti.com](#) または TI 製品などに関連して提供される他の適用条件に従い提供されます。TI がこれらのリソースを提供することは、適用される TI の保証または他の保証の放棄の拡大や変更を意味するものではありません。TI がカスタム、またはカスタマー仕様として明示的に指定していない限り、TI の製品は標準的なカタログに掲載される汎用機器です。

お客様がいかなる追加条項または代替条項を提案する場合も、TI はそれらに異議を唱え、拒否します。

Copyright © 2026, Texas Instruments Incorporated

最終更新日：2025 年 10 月