

Application Note

在 MSPM0 低功耗 MCU 上使用边缘 AI 来进行交流电弧故障检测



Akshat Aggarwal, Laavanaya Dhawan, Nathan Nohr, Vinamra Shrivastava

摘要

本应用手册演示了如何在具有集成神经处理单元 (NPU) 的低成本 MSPM0G5187 微控制器上，使用边缘 AI 实现实时交流电弧故障检测。该设计将 TIDA-010971 模拟前端参考设计与机器学习推理相结合，可检测住宅和商业电气系统中的串联电弧故障。目标性能包括小于 10ms 的响应时间和大于 95% 的检测精度，同时保持对屏蔽负载的抗扰性符合 UL 1699 指南。借助该参考设计，工程师能够开发符合美国国家电气规范 (NEC) 要求的电弧故障断路器 (AFCI) 产品。

商标

所有商标均为其各自所有者的财产。

内容

1 简介.....	2
2 快速入门指南.....	6
3 技术背景.....	9
4 系统要求.....	13
5 硬件设计.....	15
6 模型开发.....	16
7 固件实现.....	22
8 代码示例.....	24
9 性能基准测试.....	25
10 功耗优化.....	25
11 测试和验证.....	25
12 排查.....	26
13 总结.....	27
14 参考资料.....	28
15 术语.....	28
16 常见问题解答.....	29

1 简介

电弧故障问题

电弧故障是住宅和商业楼宇电气火灾的主要原因之一。当电流流经意外路径，产生高温等离子电弧，从而点燃周围材料时，就会发生电弧故障。根据美国消费品安全委员会 (CPSC) 和美国消防协会 (NFA) 的数据，电弧故障每年导致 3 万多起家庭火灾，导致数百人死亡，财产损失超过 10 亿美元。(来源：NFA 电气火灾统计数据，2019-2023)

表 1-1. 电弧故障类型

类型	说明	特征
串联电弧	与负载串联的高阻抗连接 (例如，导线损坏、连接松动)	电流流经电弧；负载仍在运行，但功率降低
并联电弧	通过损坏的绝缘实现火线对中性线或火线对地故障	高故障电流；在未连接负载的情况下可能发生

串联电弧和并联电弧故障的比较。串联电弧发生在电路中负载的电流路径中；并联电弧发生在火线及中性线或接地端

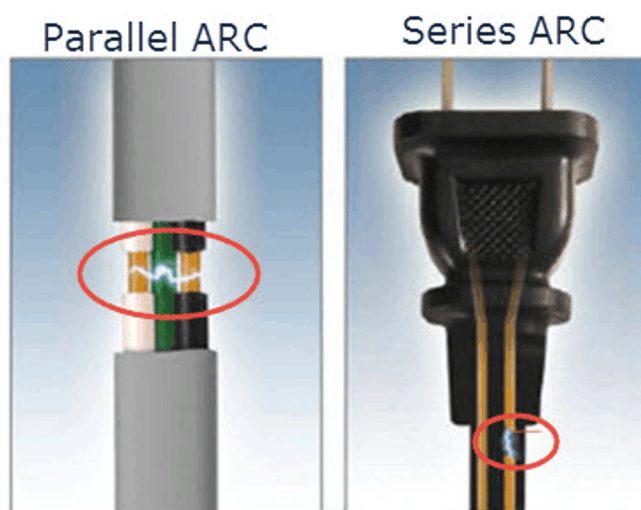


图 1-1. 串联和并联电弧故障

传统保护失败的原因

传统的断路器和保险丝可防止过流情况，但无法检测电弧故障，因为：

- 串联电弧消耗的电流可能比正常负载小
- 电弧电流波形与简单的过流特征不同
- 电弧在不受断路器监控的频率 (kHz 至 MHz) 下发生

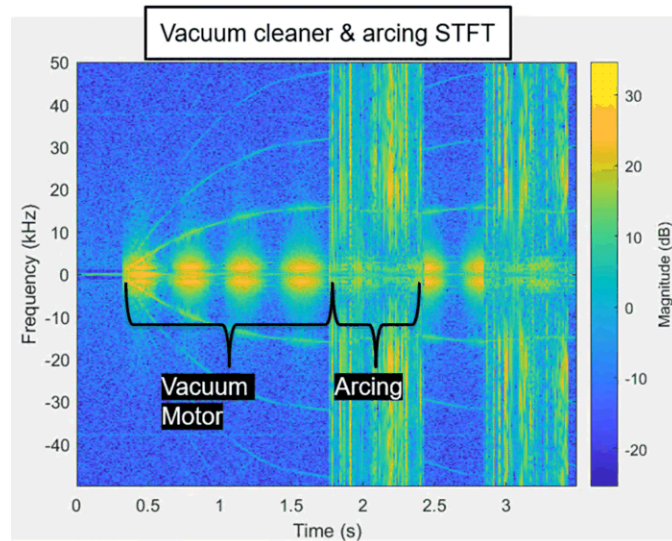


图 1-2. 电弧故障与正常负载波形

UL 1699 标准概述

UL 1699 是电弧故障断路器 (AFCI) 的安全标准，定义了住宅电路保护器件的测试要求。在北美销售的 AFCI 必须符合 UL 1699，美国国家电气规范 (NEC) 要求在大多数生活空间中保护 AFCI。

表 1-2. 关键 UL 1699 测试类别

测试类别	说明	ML 模型相关性
碳化路径电弧	通过碳化绝缘产生电弧	主要检测目标
点接触电弧	金属与金属间歇性接触	高频特征检测
屏蔽测试	正常负载模拟电弧特征	预防误报
意外跳闸	调光器、真空吸尘器、钻机、SMPS	在这些情况下，模型不得跳闸

本应用手册提供了一种参考设计来帮助工程师开发 AFCI 产品。生产器件需要通过正式的 UL 认证测试。所介绍的设计原理及测试方法符合 UL 1699 要求。

表 1-3. 设计目标与 UL 1699

Metric	UL 1699 要求	此设计目标
串联电弧检测	必需	是
并联电弧检测	必需	否
响应时间	因测试而异	<10ms
屏蔽负载抗扰度	不得妨碍跳闸	>99% 特异性

为何使用边缘 AI 进行电弧故障检测？

传统的电弧故障检测依赖于用于监控特定频带或波形特征的模拟电路。这些方法虽然有效，但存在以下局限性：

- 固定阈值无法适应不同的负载条件
- 有限的特性提取遗漏了复杂的圆弧特征
- 高误报率，具有挑战性的负载（调光器，电机）

边缘 AI 支持基本不同的方法：

1. 复杂模式识别：神经网络可以学习电弧特征与正常负载噪声之间的细微差异
2. 多特性分析：将时域、频域和电能特性相结合
3. 自适应检测：在不同数据集上训练的模型比固定规则更具普遍性
4. 设备上推理：无需云连接即可进行实时检测

表 1-4. 边缘 AI 与传统模拟 AFCI 比较

方面	传统模拟 AFCI	边缘 AI AFCI (此设计)
检测方法	固定模拟滤波器 + 比较器	带有学习特性的 ML 模型
特征提取	硬件滤波器（限制频带）	软件 FFT（全频谱分析）
阈值自适应	固定或手动调优	从训练数据中获得
误报处理	附加模拟滤波器	ML 模型在掩膜负载上训练
更新功能	所需硬件更改	可进行固件/模型更新
开发工作量	模拟电路设计专门知识	ML 训练 + 嵌入式部署
BOM 成本	多个模拟 IC	单个 MCU，支持 NPU
功耗	可变（通常更高）	低功率 MCU
灵活性	应用特定	能够适应不同场景

为何选择 MSPM0？

MSPM0G5187 特别适用于边缘 AI 电弧故障检测：

表 1-5. MSPM0G5187 特性

特性	电弧检测的优势
TinyEngine NPU	硬件加速的神经网络推理
80MHz Cortex-M0+	用于实时特征提取的快速 DSP
12 位 1.6Mps ADC	高速采样可捕获电弧特征
128KB 闪存	足以处理 ML 模型和应用代码
32KB SRAM	足够的张量空间进行推理
低功耗	使用低功耗 MCU 实现常开型监控
具有成本效益	低 BOM 成本，适用于消费类 AFCI 产品
小型封装	4mm×4mm 选项，可实现紧凑型设计



不具有 NPU 的其他 MSPM0 型号可以使用仅软件实现来运行边缘 AI 推理，但不具备硬件加速。

2 快速入门指南

本节提供了运行电弧故障检测演示的最快路径。有关详细说明，请参阅后续章节。

要求

您需要进行的准备

硬件：

- [LP-MSPM0G5187 LaunchPad](#)
- [TIDA-010971 模拟前端电路板](#)
- USB Type-C 电缆
- USB 隔离器 (交流测试所需)

软件：

- [Code Composer Studio 12.x 或更高版本](#)
- [MSPM0 SDK 2.08.00 或更高版本](#)

快速入门步骤

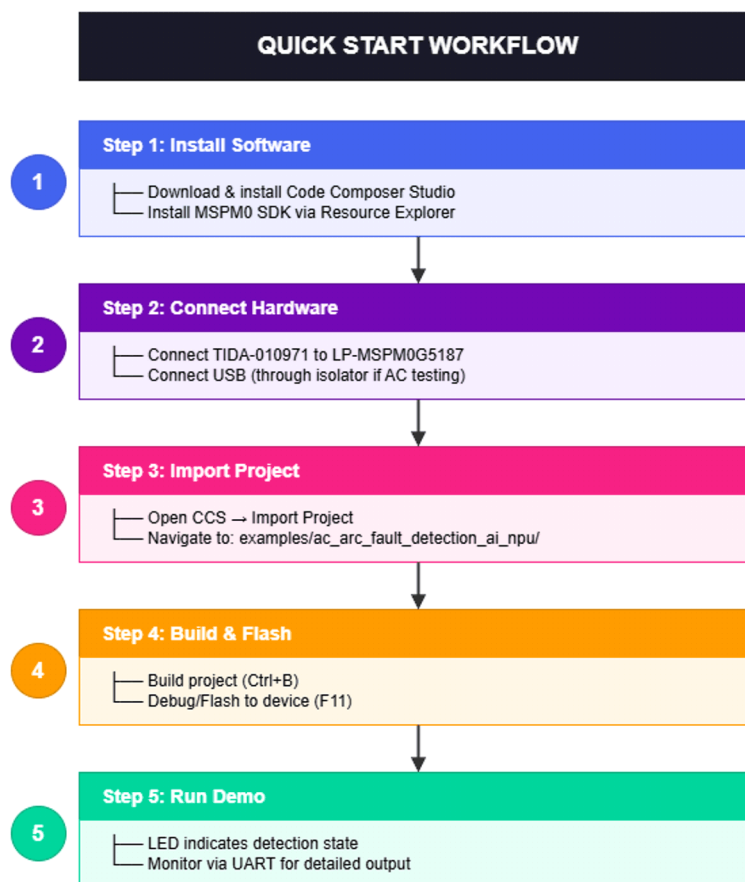


图 2-1. 快速入门工作流程，其中显示了从软件安装到运行演示的五个主要步骤

分步说明

第 1 步：安装软件

- 从 [ti.com/ccs](https://www.ti.com/ccs) 下载 Code Composer Studio
- 在安装过程中，选择“MSPM0 微控制器”
- 启动 CCS 并且打开 Resource Explorer (View→Resource Explorer)
- 搜索“MSPM0 SDK”并安装版本 2.10.00 或更高版本

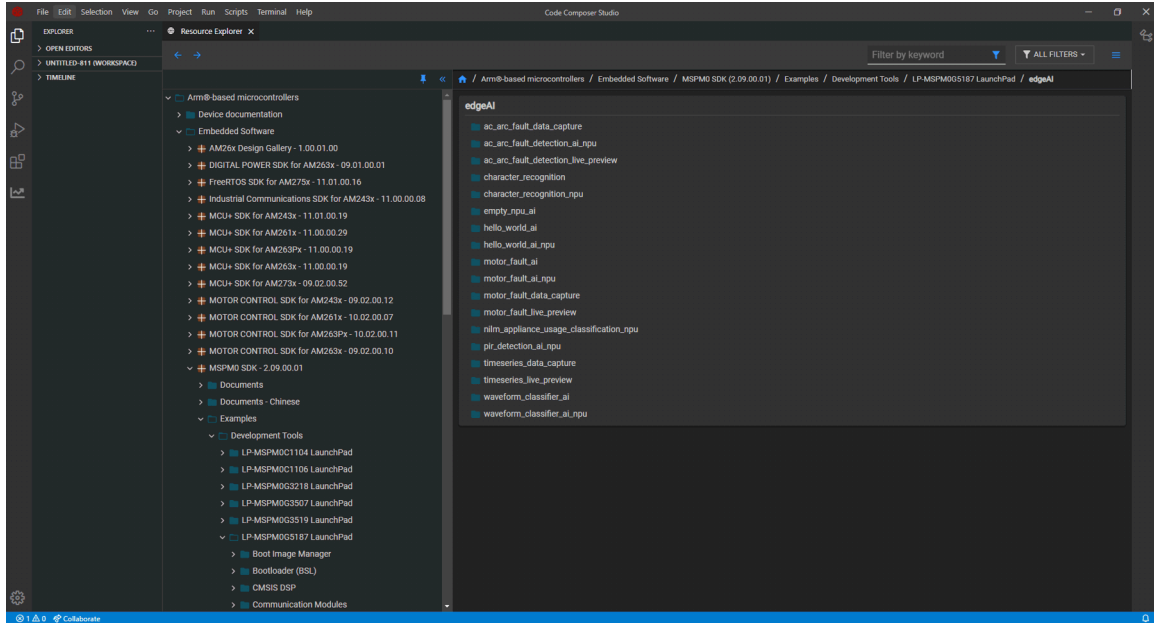


图 2-2. 显示 MSPM0 SDK 安装情况的 Code Composer Studio Resource Explorer

第 2 步：连接硬件

- 将 TIDA-010971 板连接到 LP-MSPM0G5187 LaunchPad 接头
- 验证引脚对齐情况（有关引脚排列，请参阅第 3.1 节）
- 将 USB Type-C 电缆连接至 LaunchPad

警告

使用交流电源电压进行测试时，请始终在 LaunchPad 和计算机之间使用 USB 隔离器。TIDA-010971 在危险电压下工作。

连接到 LP-MSPM0G5187 LaunchPad 的 TIDA-010971 板。注意正确的接头对齐及 USB 隔离器连接。

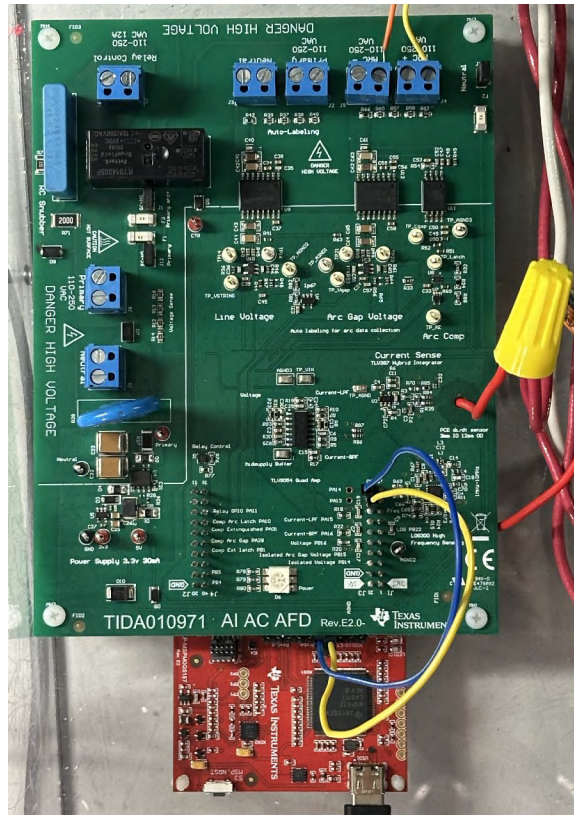


图 2-3. 连接到 LP-MSPM0G5187 LaunchPad 的 TIDA-010971 板

第 3 步：导入工程

- 在 CCS 中，选择 File → Import → CCS Projects
- 浏览至 SDK 安装：<SDK_PATH>/examples/ac_arc_fault_detection_ai_npu/LP_MSPM0G5187/
- 选择工程并点击“Finish”

第 4 步：构建和闪存

- 右键点击工程 → Build Project (或 Ctrl+B)
- 验证构建完成且没有错误
- 点击 Debug (F11) 进行闪存并开始调试
- 点击 Resume (F8) 以运行应用程序

第 5 步：验证操作

表 2-1. LED 状态表

LED 状态	含义
红色 ON	检测到电弧故障
两者均为 OFF	系统正在运行，未检测到电弧

后续步骤

运行演示之后：

表 2-2. 后续步骤

目标	部分
训练您自己的模型	[第 6 节：模型开发](#6-model-development)
修改固件	[第 7 节：固件实现](#7-firmware-implementation)
优化功耗	[第 10 节：功耗优化](#10-power-optimization)
调试问题	[第 12 节：排查](#12-troubleshooting)

3 技术背景

电弧故障电气特征

电弧故障会产生与正常负载运行不同的独特电气特征：

时域特征：

- 过零处的电流不连续（电弧在电压下降时消失）
- 叠加高频噪声的不规则电流波形
- 电流幅度可能低于正常负载（串联电弧）
- 肩部现象：电流肩靠近过零处，作为电弧重新点火

频域特征：

- 从 DC 到几十 MHz 的宽带噪声
- 时变粉红噪声频谱（ $1/f$ 滚降）
- 宽频率范围内存在电弧能量
- 与正常开关瞬态相比具有明显的频谱特征

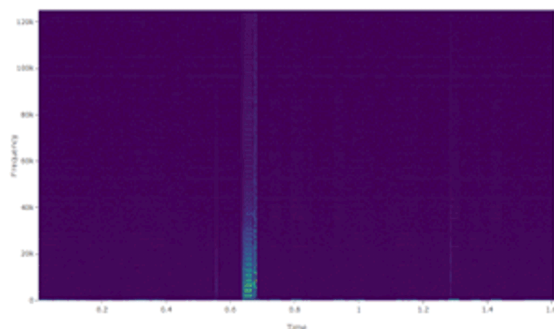


图 3-1. 正常频谱

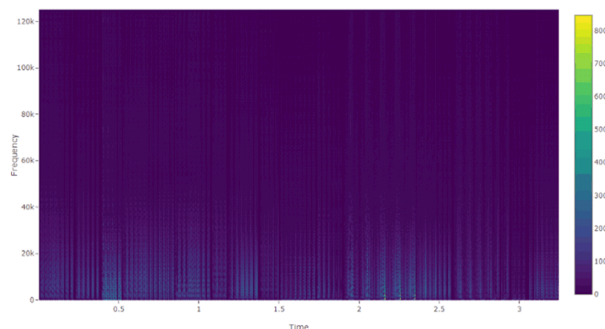


图 3-2. 电弧故障频谱

表 3-1. 电弧特征与负载噪声间的关系

特征	电弧故障	正常负载（电机/调光器）
时间模式	随机、混乱	定期、可预测
过零行为	熄灭/重新点燃	平稳过渡
能量分配	在整个光谱范围内均匀	谐波处集中

信号处理概述 (TIDA-010971)

TIDA-010971 使用带宽大于 40MHz 的 PCB Rogowski 线圈来实现多通道检测策略：

表 3-2. 信号通道

通道	频率范围	用途	ML 特性用法
带通电流	5-50kHz	可闻/超声波范围内的电弧噪声	FFT 功能的主输入
LOG300 HF 输出	1-10MHz	高频电弧能量	电弧能量幅度
低通电流	直流-60Hz	基本电流幅值	负载电平检测
线路电压	60Hz	过零基准	相位同步

关键设计概念

有关完整的电路原理图、元件值和设计计算，请参阅 [TIDA-010971 设计指南](#)。

- PCB 罗氏线圈：差分线圈输出 di/dt 信号；自谐振 >40MHz 可实现宽带电弧检测
- 混合积分器：基于 TLV387 的电路将 di/dt 转换为增益为 68.6dB 的电流比例信号
- LOG300 检测器：为 MHz 频率电弧能量检测提供 98dB 动态范围

特性提取流水线

固件从原始 ADC 样本中提取频域特性：

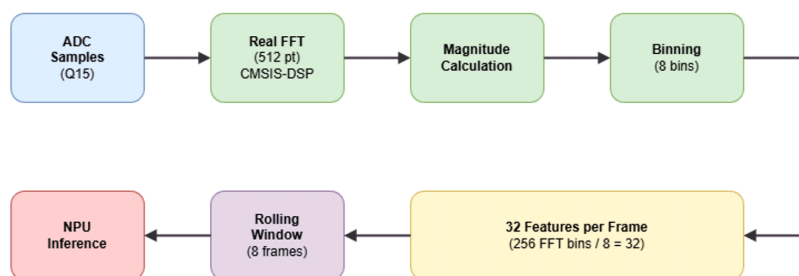


图 3-3. 特性提取流水线

时序关系

- ADC 采样间隔：9.3μs (1/107kHz 时约为 9.3μs)
- 帧周期：512 个样本 × 9.3μs 约等于 4.76ms
- 滚动窗口：8 帧 × 4.76ms 约等于 38ms 的时间上下文
- 推理速率：每帧一次 (大约 4.76ms)

配置参数 (user_input_config.h)：

备注

滚动窗口 注意事项：此处的“窗口”一词是指八个连续帧的时间滑动窗口，而不是 FFT 窗口函数（例如 Hanning、Hamming）。每个新帧都会将窗口向前移动，为电弧检测提供连续的时间上下文。

```
#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
```

处理步骤：

- 实数 FFT：使用 ARM CMSIS-DSP 的 512 点 FFT (arm_rfft_q15)
- 复数幅度：计算每个频率区间的幅度
- 直流移除：跳过直流分量 (FE_BIN_OFFSET = 1)
- 区间划分：平均 8 个相邻 FFT 频段 → 32 个特性
- 标准化：缩放至 INT8 范围 (-128 至 127)
- 帧连接：在滚动窗口中堆叠八个帧 → 总共 256 个特性

处理步骤

ML 模型架构

为了满足 MSPM0 类器件的各种存储器、延迟和精度要求，电弧故障检测解决方案提供了一系列轻量级卷积神经网络 (CNN) 模型。这些模型共享一个通用的三层卷积主干网，并且仅在通道深度上有所不同，从而实现从超低占用空间到更高精度配置的可扩展部署。神经网络将提取的特性归类为电弧故障或正常。

表 3-3. 模型架构比较

注意：下面的性能指标是针对输入尺寸 N=1、C=1、H=256、W=1 测量的

模型名称	参数	闪存大小 (B)	SRAM 大小 (B)	推理时间	精度	建议
ArcFault_model_200_t	约 200	3652	900	197.82us	99.60%	最简单、最小和最 快的模型。
ArcFault_model_300_t	约 300	3864	1644	246.91 us	99.60%	比 700 和 1400 型 号更快，但也能处 理不太复杂的数据。
ArcFault_model_700_t	约 800	4496	1644	288.36 us	99.42%	大型模型、推理速 度及内存占用之间 的最佳点。
ArcFault_model_1400_t	约 1600	5564	2408	713.31 us	99.88%	最准确的模型，用 于复杂的数据场景 (推荐)

上述电弧故障模型是专有的，因此仅在基于 GUI 的 EdgeAI Studio 中提供。

建议用于电弧故障检测的 1D CNN 架构 (电弧故障模型 1400t)

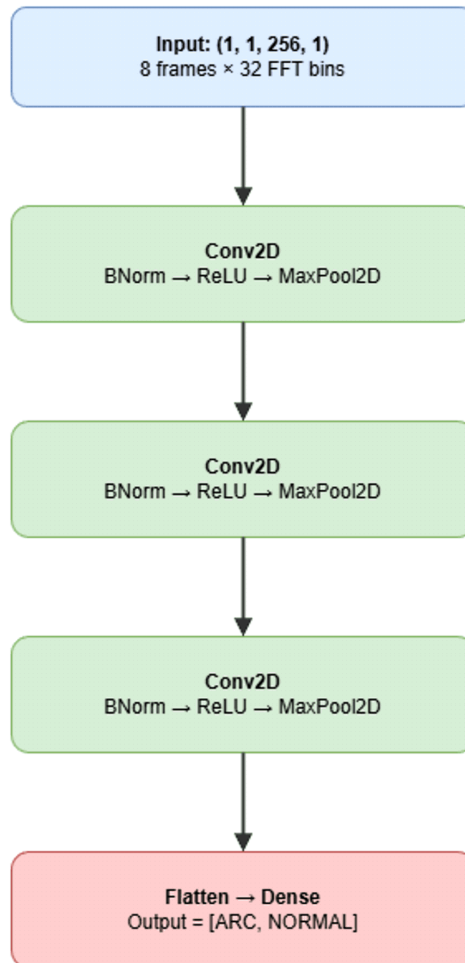


图 3-4. 建议用于电弧故障检测的 1D CNN 架构

NPU 加速

MSPM0G5187 包含一个集成的 TinyEngine™ NPU，可加速神经网络推理：

NPU 特性：

- 常见 NN 操作的硬件加速
- 用于模型存储的指令和参数存储器
- 中断驱动的完成信号
- 通过 TI 神经网络编译器 (TI-NNC) 进行 TVM 运行时集成

与 TI-NNC 集成

TI 神经网络编译器生成：

- model.a：编译的模型库
- tvmgcn_default.h：C 接口接头
- user_input_config.h：特性提取配置

4 系统要求

硬件要求

表 4-1. 硬件要求

元件	器件型号	说明
MCU/LaunchPad	LP-MSPM0G5187	MSPM0G5187 LaunchPad 开发套件
USB 电缆	USB Type-C	用于编程和调试
USB 隔离器	任何 USB 2.0 隔离器	用于确保交流测试期间的安全

主要元件

TIDA-010971 规格

表 4-2. TIDA-010971 规格

参数	值
功耗	7-17mA
罗氏线圈带宽	\50Hz 至 40MHz
罗氏线圈灵敏度	6kHz 时 ~0.4 mV/A
带通滤波器	5kHz 至 50kHz
LOG300 HF 检测	1MHz 至 10MHz
最大电流检测	320A (无削波)

LaunchPad 引脚连接

表 4-3. LaunchPad 引脚连接

引脚	ADC 通道	信号描述
PA15	ADC0_15	低通滤波电流 (60Hz 基波)
PA21	ADC0_24	LOG300 输出 (HF 电弧能量)
PB14	ADC0_21	隔离式线路电压 (AMC3330)
PA31	GPIO	ARC 指示引脚 (设置 ARC、未设置 - 无 ARC)
PA11	GPIO	继电器控制输出
PB4	GPIO	红色 LED 指示灯
PB5	GPIO	绿色 LED 指示灯

TIDA-010971 主要元件

TIDA-010971 使用专用的模拟元件，包括用于混合积分器的 TLV387 零漂移运算放大器、用于带通和低通滤波的 TLV9054 四通道运算放大器、LOG300 高频对数检测器 (98dB 动态范围)、AMC3330 隔离放大器和用于电源的 UCC28881 离线降压转换器。

有关完整的 IC 规格、电路原理图和设计文件，请参阅 [TIDA-010971 设计指南](#)

软件要求

表 4-4. 软件要求

软件	版本	用途
MSPM0 SDK	2.10.00 或更高版本	器件驱动程序和库
TI Edge AI Studio	1.6.0 或更高版本	模型训练和部署 GUI 工具
模型制作器	R1.3 或更高版本	模型训练和部署 CLI 工具
ARM CMSIS-DSP	包含在 SDK 中	FFT 和 DSP 函数
Python	3.10 或更高版本	模型训练/数据标记脚本

模型要求

表 4-5. 模型要求

参数	规格
输入尺寸	8 帧 × 32 个 FFT 区间 = 总共 256 个功能
输出形状	(1, 2) — [arc_confidence, normal_confidence]
量化	INT8 (强制 — Cortex-M0+ 上无 FPU)
模型大小	大约 14KB 参数
张量空间	大约 8KB SRAM

Tensor 形状说明

- 批次 (1)：一次性单一推理
- 通道 (1)：单输入通道 (灰度式表示)
- 高度 (1024)：连接的特征矢量 (8 帧 × 32 个特征)
- 宽度 (1)：单个元素宽度 (矢量表示为 2D)

5 硬件设计

系统概述

完整的电弧故障检测系统包括连接到 LP-MSPM0G5187 LaunchPad 的 TIDA-010971 模拟前端：

完整的系统方框图显示了连接至 LP-MSPM0G5187 以进行边缘 AI 电弧故障检测的 TIDA-010971 模拟前端

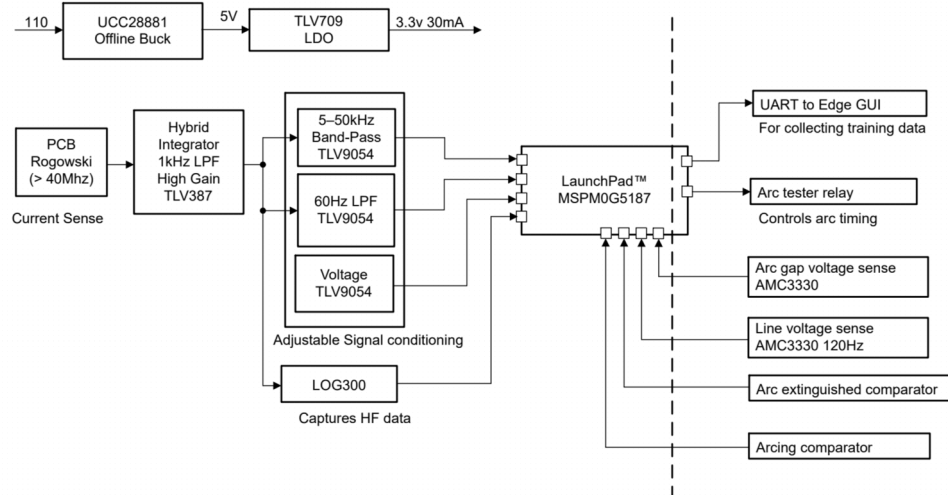


图 5-1. TIDA-010971 系统方框图

主要元件

- TIDA-010971：提供模拟信号调节、电源及隔离
- LP-MSPM0G5187：执行 ADC 采样、特征提取、NPU 推理和跳闸决策

有关完整的系统原理图、PCB 布局和设计文件，请参阅 TIDA-010971 设计指南。

硬件设置

电路板连接

使用 BoosterPack 接头将 TIDA-010971 连接到 LP-MSPM0G5187。上电之前验证引脚对齐。将电弧发生器设置与模拟前端连接，用于生成不同的电弧或正常情况。

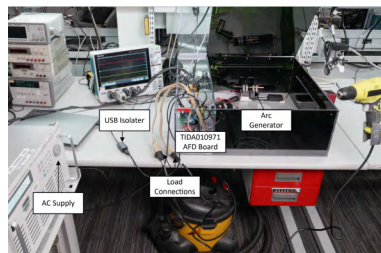


图 5-2. 电弧发生器测试装置

安全注意事项

警告

危险电压

此设计在交流电源电压 (110-250 VAC) 下工作，可能导致严重伤害或死亡。请遵循所有安全准则：

1. 需要 USB 隔离：在交流测试期间，务必在 LaunchPad 和 PC 之间使用 USB 隔离器
2. 外壳：仅在正确接地的绝缘外壳中运行
3. 电流限制：在开发过程中使用限流交流电源
4. 无触摸规则：通电时请勿触摸电路板或接头
5. 放电时间：在处理前让电容器放电 (>5 分钟)
6. 合格人员：只有合格的电气工程师才能进行交流测试

6 模型开发

本部分为使用 TI 的 Edge AI 工具开发和部署电弧故障检测 ML 模型提供了指南。

开发工作流程概述

从数据收集到部署的端到端模型开发工作流程

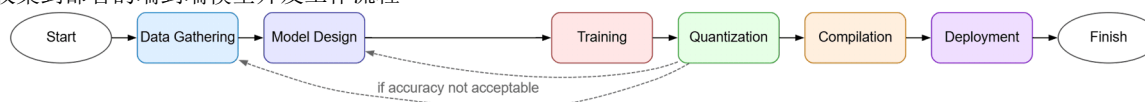


图 6-1. 模型开发工作流程

数据搜集

硬件设置

- 刷写数据捕获固件：examples/ac_arc_fault_data_capture/LP_MSPM0G5187/
- 通过 BoosterPack 接头将 TIDA-010971 连接到 LP-MSPM0G5187
- 连接 USB (通过隔离器进行交流测试)
- 串行连接：5,820,000 波特、二进制帧

数据收集方案

在受控条件下收集每个类别的数据。每种负载类型包含 20000+ 半周期捕获的最小值 (在正常配置和电弧配置中大约平均分配三分钟)。

表 6-1. 等级 1：正常运行 (无跳闸)

负载类型	说明	捕获
电感	交流继电器、压缩机电机、真空、车间工具	10000+
LED	卤素照明、600W — 1000W、带调光器	10000+
SMPS	600W AC/DC	10000+
混合	5A 电阻负载下的上述组合	10000+

表 6-2. 等级 2：电弧故障屏蔽负载

电弧类型	说明	捕获
配置 A	加载串联的电弧发生器	10000+
配置 B	使用串联和并联的 5A 电阻负载的电弧发生器负载	10000+
配置 C	使用电弧发生器将负载与 5A 电阻负载并联	10000+
配置 D	均使用电弧发生器的负载和 5A 电阻负载	10000+

数据标记

捕获的数据标记为两个类。标记工具提供两种可配置的模式，用于从原始波形 CSV 生成 ARC 数据和 NORMAL 训练数据：帧模式和窗口模式。这两种模式都使用基于电弧电压的启发法来确定地端实况标签，并且独立于任何机器学习模型。虽然支持这两种模式以适应不同的数据集和前端架构，但对于大多数应用而言，帧模式是推荐的默认模式，因为它能够更好地保持事件连续性并产生更清晰的类边界。

帧模式 (推荐)

用途：帧模式旨在捕获自然的电弧事件持续时间，并产生可变长度的 ARC 和可紧密反映实际电气行为的 NORMAL 段。它特别适用于基于 LOG300 的数据集和保持时间连续性很重要的场景。

工作原理：

1. 波形分为固定长度帧
2. 对于每个帧，会根据可配置的阈值评估电弧间隙电压统计数据（例如峰值和 RMS）
3. 在连续帧中应用时间平滑处理，以抑制隔离的尖峰和开关瞬变。基于状态机的合并将连续的 ARC 或 NORMAL 帧合并成较长的段。将丢弃低于最短持续时间的片段，以避免对短而嘈杂的片段进行训练。将重叠已确认 ARC 区域的帧从 NORMAL 输出中排除，以验证纯正常样本

输出特征

1. 可变长度的 ARC 和 NORMAL 分段
2. 强大的噪声抑制和更低的标签抖动
3. 保留电弧事件的自然起始/终止边界
4. 可配置的保存通道（例如 LOG300 包络或电流），而标记逻辑始终使用电压。

使用场合：

1. LOG300 或基于包络的前端
2. 需要高标签保真度和事件连续性的应用
3. 为瞬态噪声或边界伪影敏感的模式创建数据集

窗口模式 (推荐)

用途：窗口模式使用滑动窗口方法生成固定长度的训练样本。当需要统一的输入大小或为需要恒定长度特性的架构构建数据集时，这很有用。

工作原理：

1. 该信号首先被划分为多个帧。
2. N 个连续帧的窗口在波形中滑动。
3. 在每个窗口内，会对类似电弧的帧数量（基于电压阈值）进行计数。
4. 如果计数超过可配置的阈值，则整个窗口被标记为 **ARC**；如果电压电平足够低且稳定，则标记为 **NORMAL**；否则，该窗口将被丢弃。
5. 提取的与标记窗口相对应的波形将保存为固定大小样本。

输出特征：

1. 固定长度的 **ARC** 和 **NORMAL** 窗口
2. 更轻松的批处理和一致的模型输入尺寸
3. 可能会丢弃模糊窗口以保持可信度
4. 标记仍由电压驱动；保存的通道通常为最新通道

使用场合

1. 需要统一样本长度的流水线。
2. 严格的批处理简单性超过事件连续性要求的情况。

建议

这两种模式受到完全支持，并共用基于电压的相同接地真值逻辑。但是，建议将帧模式用作默认模式，因为它：- 更好地保持现实世界的电弧事件连续性 — 通过时间平滑减少噪声引起的假标签，并且 — 通过明确排除电弧相邻区域产生更清洁的 **NORMAL** 样本。当固定大小输入或快速数据集标准化是主要要求时，窗口模式仍然很有用。

有关更多信息，请参阅 `Tinyml-modelzoo->examples->ac_arc_fault->readme.md`

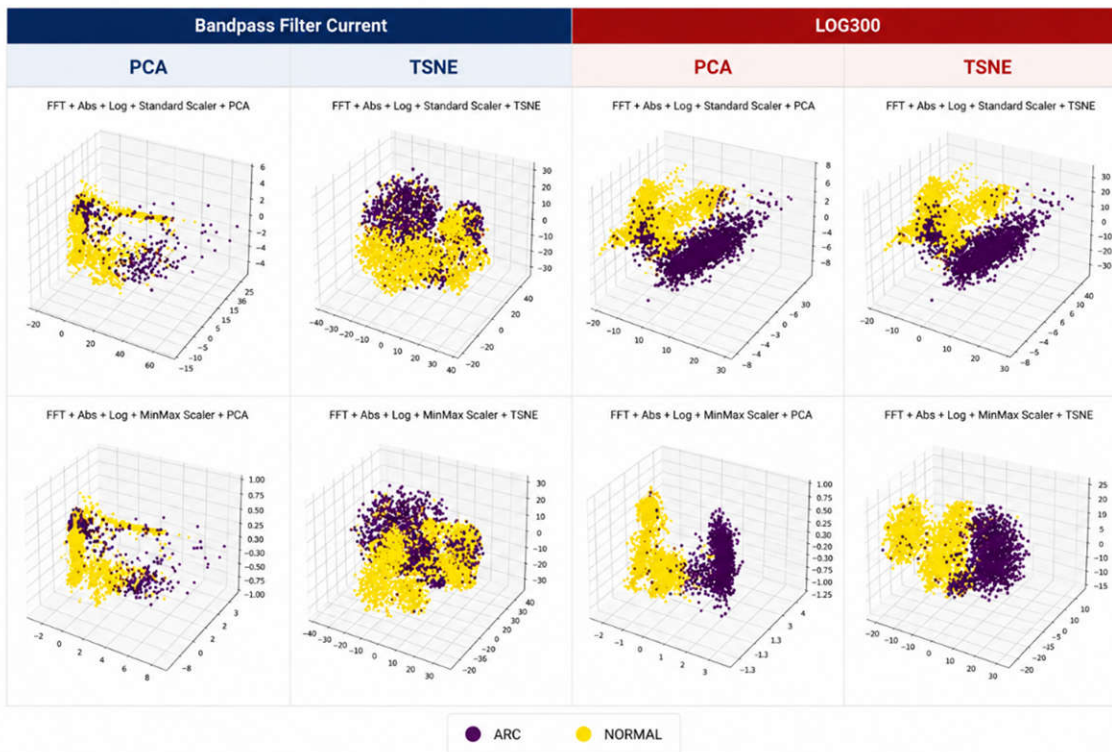
数据集分析和选择

拟合优度 (GOF) 图解用作早期评估，用于在模型训练之前评估数据集中的特性分离性。这些图帮助比较不同预处理组合中的 **ARC** 和 **NORMAL** 样本的分布，包括 FFT、小波变换、缩放、PCA 和 t-SNE。GOF 图中的强分级分离表明，所选的信号链和特性提取方法适用于分级，而类别之间的重叠表明可能需要进一步调优。

对于带通滤波电流数据集，GOF 图显示明显的 **ARC/NORMAL** 重叠，尤其是基于 FFT 的预处理，表明一些与电弧相关的特征被屏蔽。相比之下，基于 LOG300 的数据集产生了更清晰的类群集，这表明功能可分离性得到改

善，并且更适合下游模型训练。有关数据集时基故障良率、特性提取质量、GOF 图和文件级分类摘要的更多详细信息，请参阅[高级特性](#)下的 TI Tiny ML Tensorlab 用户指南。

带通滤波电流和 LOG300 数据集的 GOF 比较



模型架构

而第 4.4 节中提到的电弧故障模型是交流电弧故障检测的推荐模型选择。还可使用多种通用时间序列模型进行分类。

TI Edge AI Studio (GUI) 和 TensorLab (CLI) 均提供通用时间序列模型模板，可快速开发和评估嵌入式 ML 应用。这些模板提供即用型 NPU 兼容架构，这些架构可以用非常少的配置工作就进行自定义数据集训练。

其中一种参考架构是 CLS_6K_NPU 模型，这是一种轻量级可分离卷积神经网络。此模型以及一系列其他参考分类架构可以在 TI Edge AI Studio (GUI) 和通过 CLI 工具提供的 Model Zoo 中获取。

```
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

```

备注

此示例使用支持 PyTorch 框架的 TI Edge AI Studio GUI/CLI 工具

模型训练

表 6-3. ArcFault_model_1400_t 的推荐超参数：

参数	值
Batch size	50
Learning rate	0.04
Optimizer	SGD
Weight decay	1e-5
Dataset split	60% 训练、30% 间隔、10% 测试

训练选项

- GUI：使用 **CCStudio™ Edge AI Studio GUI** 进行指导型训练工作流程
- CLI：使用 **tinymml-tensorlab** 进行命令行训练

启用定量感知训练 (QAT)，以获得最佳的 INT8 精度。Tinymml-tensorlab 和 Edge AI Studio 内置对 8 位 NPU 感知量化的支持。

量化

为什么要进行 INT8 量化？

MSPM0G5187 Cortex-M0+ 没有硬件浮点单元：

- FP32 操作需要软件仿真（非常慢）
- INT8 运算在单个周期内执行
- NPU 仅支持 INT8 运算

- 模型大小减小 ~4 倍

量化感知训练 (QAT)

QAT 在训练期间模拟量化，以实现出色的精度保持（通常下降 <1%）：

```
import torch.quantization as quant

# Prepare model for QAT
model.qconfig = quant.get_default_qat_qconfig('fbgemv')
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)
```

输入标准化

归一化参数特定于模型，并由 TI-NNC 生成：

```
// 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
```

重要提示：始终使用编译期间生成的模型 `user_input_config.h` 文件中的值。

使用 TI-NNC 进行模型编译

TI 神经网络编译器 (TI-NNC) 对经过训练的“onnx”INT8 模型进行转换，以便于 MSPM0 NPU 执行。

编译生成：

- `model.a` — 编译后的模型库 (~14 KB)
- `tvmgen_default.h` — C 接口接头
- `user_input_config.h` — 特性提取配置

集成到 CCS 工程：

- 将生成的文件复制到 `project/model/` 目录
- 将 `model.a` 添加到链接器输入文件
- 在源代码中包含标头：`#include model/tvmgen_default.h`

有关安装、CLI 语法和详细编译说明，请参阅 TI-NNC 用户指南。

7 固件实现

本节详细介绍了 LP-MSPM0G5187 上用于实时电弧故障检测的固件架构。

系统架构

主环路状态机：

显示主应用程序循环和中断处理程序的固件状态机图

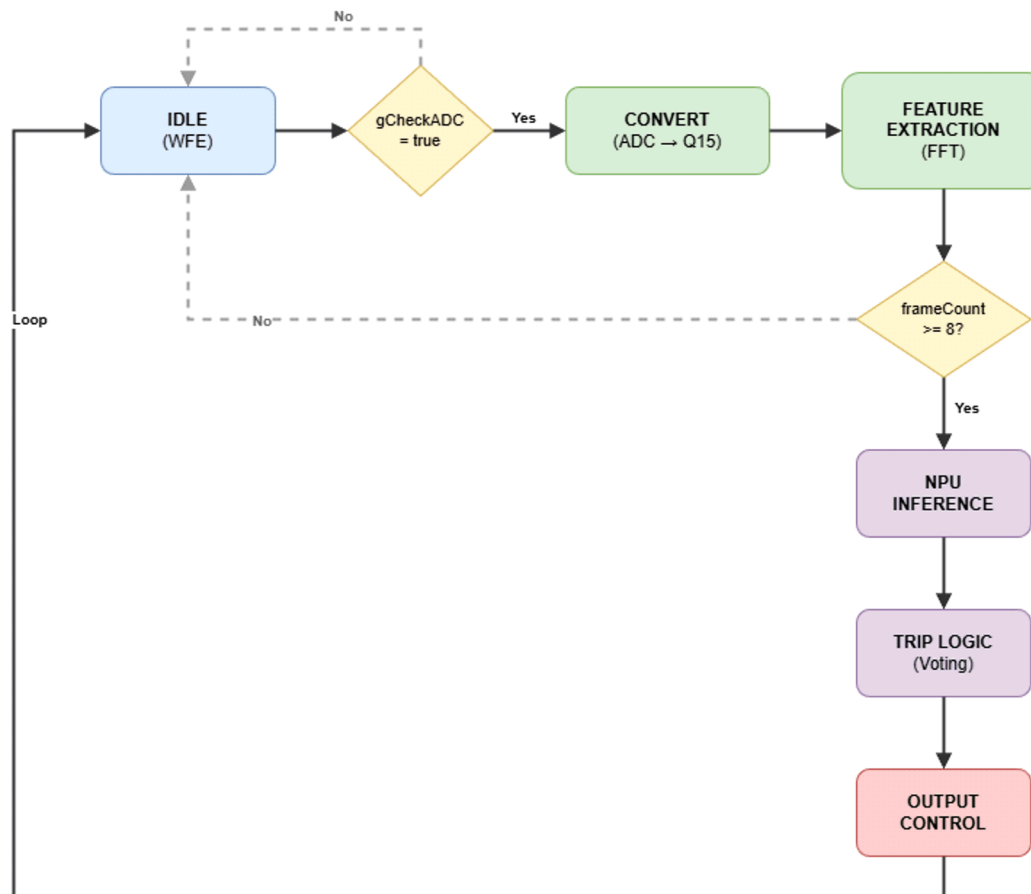


图 7-1. 固件状态机图

外设配置

使用 TI SysConfig 配置 MSPM0G5187 外设。密钥设置：

表 7-1. 外设配置

外设	配置
ADC12_0	12 位，SYSOSC 时钟，启用 DMA，通道 ADC0_14
DMA	乒乓传输，每个缓冲区 512 个样本
GPIO	PA11 (继电器)，PB4/PB5 (LED)，PA31 (比较器输入)

存储器映射

表 7-2. 存储器映射

闪存存储器 (128KB)	SRAM (32KB)
可用 (约 94KB)	可用 (约 23KB)
模型 (约 5.5 KB)	栈 (约 2KB)
模型 (约 5.5 KB)	张量空间 (2 KB)
应用程序代码 (约 20KB)	功能 Buf (1 KB)
	ADC 缓冲器 (1KB)
	FFT 暂存 (2KB)

ADC 数据采集

乒乓缓冲器策略

```
/* 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
```

DMA 填充一个缓冲区，同时 CPU 处理另一个缓冲区，从而实现连续采样。

Log 300 数据扩展

```
/* 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
    }
}
```

特征提取

```
#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 推理

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

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

/* Run inference */
tvmgen_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 */
```

跳闸逻辑 (表决算法)

```
#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 代码示例

现有示例

表 8-1.

示例	位置	说明
电弧故障检测 (NPU)	`examples/ac_arc_fault_detection_ai_npu/'	完整的推理应用程序
数据采集	`examples/ac_arc_fault_data_capture/'	训练数据收集
实时预览	`examples/ ac_arc_fault_detection_live_preview/'	实时可视化

工程文件结构

```
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
│   ├── tvmggen_default.h               # Model interface
│   └── user_input_config.h              # Feature config
```

图 8-1. 工程文件结构

9 性能基准测试

备注

下面给出的性能基准适用于输入配置为 [1,1,256,1] 的 ArcFault_model_1400_t。

表 9-1. 性能基准测试

Metric	值	注释
模型大小 — ArcFault (闪存)	大约 5564B	编译的 model.a
应用代码	约 13KB	包括库
张量空间 (SRAM)	约 2KB	运行时工作区
使用的总 SRAM	约 9KB	32 KB 中可用量
时序		
ADC 采样	4.75ms	107kSps 时 512 个样本
特征提取	大约 5ms	FFT + 区间划分
NPU 推理	大约 0.7ms	取决于具体模型
总帧时间	大约 7.19ms	每个推理周期
端到端延迟	<150ms	包括 8 帧表决
电源		
有功功率 (MCU)	约为 8mW	@ 80 MHz , 3.3 V
系统总计	约为 55mW	包括 TIDA-010971 AFE
精度		
检测精度	>99%	经过典型训练的模型
误报率	0.01%	错误标记的负值百分比
精密	99.97%	正确预测的正值百分比。
回顾	99.72%	捕获的实际正值百分比
F1 分数	99.84%	实现精度与回调的平衡。

10 功耗优化

该参考设计使用持续监控实现超低延迟，非常适合安全关键型 AFCI 应用：

- ADC 以 107kSps 的速度连续运行
- NPU 推理大约每 9.3ms 一次
- 常开运行

对于电池供电的应用，占空比可以将功耗降低约 50%，但可以按比例增加检测延迟。不建议用于满足 UL 1699 合规性。

11 测试和验证

所需设备

表 11-1. 所需设备

设备	用途
示波器	波形捕获 (≥100MHz)
电流探针	电流测量 (直流-50MHz)
USB 隔离器	PC 保护 (2500V 隔离)

安全警告

警告

电弧生成安全：电弧生成涉及极端危险，包括触电、严重烧伤、火灾和有毒烟雾。

只有具备适当 PPE 的合格电气工程师才能进行电弧测试。在安全观察员在场的情况下，在防火环境中工作。使用限流交流电源。

基本测试过程

- 上电：验证 3.3V 电源轨，确认 MCU 引导 (LED 指示)。注意：使用 Launchpad 时，请勿从其他输入为电路板供电。
- 正常运行：施加各种负载，确保没有误跳闸 (每 5 分钟)
- 电弧检测：生成受控电弧，确认在 100ms 内检测
- 屏蔽负载：测试调光器、真空吸尘器、钻机 — 确认不跳闸

12 排查

构建错误

表 12-1. 构建错误

错误	设计
`找不到 -lmodel`	验证 model/ directory 中的 model.a，检查链接器路径

运行时问题

表 12-2. 运行时问题

症状	设计
无论输入如何，输出都是相同的	验证标准化参数匹配训练，检查 Q15 转换

检测问题

表 12-3. 检测问题

问题	设计
缺失电弧检测	降低投票阈值 (4/8)，检查传感器带宽

备注

只有 MSPM0G5187 具有集成 NPU。其他 MSPM0 型号无法使用 NPU 加速推理。

13 总结

本应用手册介绍了一个完整的参考设计，用于在具有集成神经处理单元 (NPU) 的低成本 MSPM0G5187 微控制器上使用边缘 AI 实现交流电弧故障检测。该设计解决了一个关键的安全挑战：电弧故障每年导致 30,000 多起家庭火灾，但传统的断路器和保险丝无法检测到这些故障，因为串联电弧可能比正常负载消耗更少的电流，并在热保护器件不监测的频率 (kHz 至 MHz) 下工作。

该设计方案利用机器学习推理来克服固定模拟电路的局限性。通过将硬件加速的神经网络处理与 TIDA-010971 模拟前端的复杂信号调节相结合，该系统可实现 >99% 的检测精度，<10ms 的响应时间和 >99% 的特异性来屏蔽负载，符合 UL 1699 安全标准。该架构采用四系列 CNN 模型集，推荐的 ArcFault_model_1400_t 可提供 99.88% 的精度，而仅消耗 5.6KB 闪存和 2.4KB SRAM。

固件实现使用 107kSps 的连续 ADC 采样和乒乓 DMA 传输来实现不间断数据采集。通过 512 点 FFT 进行实时特征提取，每个推理周期可生成 256 个时间频率特征，由 NPU 在大约 0.7ms 内处理。多数表决算法 (8 帧中的 5 个) 可确保可靠检测，同时更大限度地减少干扰性跳变。总端到端延迟保持在 150ms 以下，MCU 的有功功耗约为 8mW，整个系统的有功功耗约为 55mW。该参考设计包括有关模型开发的全面指导，涵盖从数据收集和标记策略到通过 TI 神经网络编译器进行量化、训练和部署的各种指导。支持帧模式和窗口模式标记方法，以适应不同的数据集和前端架构。借助提供的示例、固件代码和性能基准，工程师能够以最少的开发工作量开发用于住宅和商业电气系统的生产 AFCI 产品，同时保持符合美国国家电气规范 (NEC) 要求和 UL 1699 标准。虽然该参考设计为 AFCI 开发提供了坚实的基础，但量产器件需要经过正式的 UL 认证测试和验证。模块化架构和可扩展模型选项支持在各种性能和成本要求下进行部署，从超低尺寸实施到适用于复杂电气环境的高精度配置，不一而足。

14 参考资料

硬件

1. 德州仪器 (TI), [基于边缘 AI 的断路器交流电弧故障检测参考设计](#), 设计指南。
2. 德州仪器 (TI), [MSPM0G5187 评估模块](#), EVM 用户指南。
3. 德州仪器 (TI), [MSPM0G5187 具有 TinyEngine™ NPU 的混合信号微控制器](#), 数据表。

软件

1. 德州仪器 (TI), [MSPM0-SDK](#), 软件。
2. 德州仪器 (TI), [Edge AI Studio](#), 软件。
3. 德州仪器 (TI), [TI MCU 神经网络编译器用户指南 -2.1.1 LTS](#), 网页。
4. 德州仪器 (TI), [Texas Instruments/tinymt-tensorlab](#), 网页。

标准

1. UL 1699, 电弧故障断路器, 标准。
2. NEC 第 210.12 条, AFCI 保护要求, 标准。
3. IEC 62606, AFDD 的一般要求, 标准。

15 术语

表 15-1. 术语

术语	定义
AFCI	电弧故障断路器
DMA	直接存储器存取
边缘 AI	边缘器件上的机器学习推理
FFT	快速傅里叶变换
NPU	神经处理单元
Q15	定点格式, 有 15 个小数位
QAT	量化感知训练
TI-NNC	TI 神经网络编译器

16 常见问题解答

Q：我是否可以将此设计用于经过认证的 AFCI 产品？

A：这是一个参考设计。生产 AFCI 需要正式的 UL 1699 认证测试。

Q：这是否适用于其他 MSPM0 器件？

A：NPU 加速推理需要 MSPM0G5187（仅限具有集成 NPU 的 MSPM0）。其他型号可以在没有硬件加速的情况下运行仅软件推理。

Q：我需要多少训练数据？

A：每个类别至少捕获 100 次。生产模型通常每个类别使用 500 多次捕获。

Q：我可以使用不同的电流传感器吗？

A：可以。TMCS1123 霍尔效应传感器或电流互感器能够进行适当的信号调节。确保用于电弧检测的 DC-MHz 带宽。

Q：为什么推理结果总是相同？

A：在 user_input_config.h 中检查标准化参数是否匹配训练。验证 Q15 转换，并确认 model.a 与体系结构匹配。

重要通知和免责声明

TI“按原样”提供技术和可靠性数据（包括数据表）、设计资源（包括参考设计）、应用或其他设计建议、网络工具、安全信息和其他资源，不保证没有瑕疵且不做任何明示或暗示的担保，包括但不限于对适销性、与某特定用途的适用性或不侵犯任何第三方知识产权的暗示担保。

这些资源可供使用 TI 产品进行设计的熟练开发人员使用。您将自行承担以下全部责任：(1) 针对您的应用选择合适的 TI 产品，(2) 设计、验证并测试您的应用，(3) 确保您的应用满足相应标准以及任何其他安全、安保法规或其他要求。

这些资源如有变更，恕不另行通知。TI 授权您仅可将这些资源用于研发本资源所述的 TI 产品的相关应用。严禁以其他方式对这些资源进行复制或展示。您无权使用任何其他 TI 知识产权或任何第三方知识产权。对于因您对这些资源的使用而对 TI 及其代表造成的任何索赔、损害、成本、损失和债务，您将全额赔偿，TI 对此概不负责。

TI 提供的产品受 [TI 销售条款](#)、[TI 通用质量指南](#) 或 [ti.com](#) 上其他适用条款或 TI 产品随附的其他适用条款的约束。TI 提供这些资源并不会扩展或以其他方式更改 TI 针对 TI 产品发布的适用的担保或担保免责声明。除非德州仪器 (TI) 明确将某产品指定为定制产品或客户特定产品，否则其产品均为按确定价格收入目录的标准通用器件。

TI 反对并拒绝您可能提出的任何其他或不同的条款。

版权所有 © 2026，德州仪器 (TI) 公司

最后更新日期：2025 年 10 月