

## RP2350の他のエラッタ (E9を除く)

Raspberry Pi Pico 2に搭載されているRP2350マイクロコントローラ (RP2350およびRP2350A) には、E9 (GPIOプルダウン問題) 以外にも複数のエラッタが公式に報告されています。これらのエラッタは、RP2350のシリコン設計上の不具合で、特定の機能や動作に影響を与える可能性があります。以下に、E9を除くすべてのエラッタ (E1～E8、E10以降) を詳細に説明し、影響と回避策をまとめます。情報は公式のRP2350 Errataシート (<https://datasheets.raspberrypi.com/rp2350/RP2350-Errata.pdf>) を基にしています。

## RP2350エラッタ一覧 (E9を除く)

以下は、2025年10月時点で公式に記載されているE1～E8、E10以降のエラッタの概要です。エラッタはハードウェア固有の問題であり、ソフトウェアや設計で回避する必要があります。

---

### E1 - Boot ROM USB Boot Failure

- ・**概要:** USBブートモードで、特定の条件下でBoot ROMが正しく動作せず、デバイスがブートに失敗する。
- ・**影響:**
  - ・USB経由でのファームウェアアップロード（例: UF2ファイルのドラッグアンドドロップ）が失敗する可能性。
  - ・特に、USBホストとのタイミングやプロトコルエラーが発生する場合に顕著。
- ・**条件:** USBブートシーケンス中に、ホスト側が非標準的なタイミングで応答すると発生。
- ・**回避策:**
  - ・信頼性の高いUSBホスト（PCやハブ）を使用し、ケーブル品質を確認（高品質なUSBケーブル推奨）。
  - ・ソフトウェアでブートローダを再試行（リセット後再接続）。
  - ・代替として、SWD（Serial Wire Debug）経由でのブートを使用。
- ・**影響度:** 中（USBブートを多用する場合に注意。開発初期のファームウェア書き込みで影響大）。
- ・**公式推奨:** USBホストの互換性テストを実施。

## E2 - ADC Sample Timing Issue

- ・概要: ADC（アナログ・デジタル変換器）のサンプリングタイミングが不安定で、ノイズや不正確な値が出る場合がある。
- ・影響:
  - ・ADCピン（GPIO26～29）でのアナログ入力値が期待より高ノイズ（例: ±10LSBの誤差）。
  - ・高精度センサー（例: 温度センサー、電圧計測）で問題が発生。
- ・条件: 高速サンプリング（500ksps）や特定のクロック設定で顕著。
- ・回避策:
  - ・ADCサンプリングレートを下げる（例: 100ksps以下）。
  - ・ソフトウェアでオーバーサンプリング（複数回平均化）を実装:

c

✗ ≡ Ⓡ

```
uint32_t sum = 0;  
for (int i = 0; i < 16; i++) sum += adc_read();  
uint16_t avg = sum / 16; // ノイズ低減
```

- ・外部フィルタ（例: ローパスフィルタ、10nFコンデンサ）をADCピンに追加。
- ・影響度: 高（アナログセンサー使用時）。Pico 2のADC用途に注意。
- ・公式推奨: ハードウェアフィルタとソフトウェア補正を組み合わせる。

---

## E3 - PWM Wraparound Glitch

- ・**概要:** PWM（パルス幅変調）モジュールで、カウンタがラップアラウンド（上限に達してリセット）する際に短いグリッチ（ノイズパルス）が発生。

- ・**影響:**

- ・PWM出力（例: モータ制御、LED調光）で意図しない短いパルスが発生。
- ・特に高周波PWM（例: 100kHz以上）で顕著。

- ・**条件:** PWMカウンタがラップアラウンドするタイミング（周期終了時）。

- ・**回避策:**

- ・PWM周波数を下げる（例: 10kHz以下）でグリッチ影響を軽減。
- ・ソフトウェアでPWM出力を一時無効化し、ラップアラウンド時に再設定:

c                    ×     ≡     ⓧ

```
pwm_set_enabled(PWM_SLICE, false);
pwm_set_wrap(PWM_SLICE, value);
pwm_set_enabled(PWM_SLICE, true);
```

- ・外部回路でグリッチ除去（例: RCフィルタ、100Ω+100nF）。
- ・**影響度:** 中（PWMを高精度で使用する場合に注意）。
- ・**公式推奨:** 低周波PWM優先、またはハードウェアフィルタ追加。

---

## E4 - QSPI Flash Read Corruption

- ・**概要:** QSPI (Quad SPI) インターフェースでのフラッシュメモリ読み出しが、特定の条件下でデータ破損を起こす。

- ・**影響:**

- ・外部フラッシュメモリ (例: Pico 2の4MBフラッシュ) からのデータ読み出しでビットエラーが発生。
- ・ブートコードやファームウェアの実行で不安定性 (例: クラッシュ)。

- ・**条件:** 高クロック (例: 150MHz以上) や特定のフラッシュチップとのタイミング不一致。

- ・**回避策:**

- ・QSPIクロックを下げる (例: 100MHz以下)。Pico SDKで設定:

c	×	≡	⌚
---	---	---	---

```
qspi_set_clkdiv(2); // クロック分周を増加
```

- ・フラッシュチップの互換性確認 (WinbondやMacronix推奨)。
- ・ECC (エラー訂正コード) をソフトウェアで実装。

- ・**影響度:** 高 (フラッシュ依存のプロジェクトに影響大)。

- ・**公式推奨:** 低クロック設定と互換チップ使用。

---

## E5 - RISC-V Core Hazard Detection Bug

- ・**概要:** RISC-Vコア (Hazard3) の命令ハザード検出が不完全で、特定のパイプライン条件下で誤動作。

- ・**影響:**

- ・RISC-Vモードでのコード実行時に、まれに命令スキップやデータ破損。
  - ・特にタイトなループや高頻度割り込みで発生。

- ・**条件:** メモリアクセスと命令フェッチのタイミング競合。

- ・**回避策:**

- ・コンパイラオプションでパイプライン最適化を抑制 (例: -O0)。
  - ・クリティカルセクションにNOP命令を挿入:

c

✗ ≡ ⓧ

```
asm volatile ("nop; nop; nop;");
```

- ・可能ならArm Cortex-M33コアを使用 (RISC-V回避)。
- ・**影響度:** 中 (RISC-Vコア使用時のみ)。
- ・**公式推奨:** Armコア優先、またはパッチ適用 (将来のSDK更新で対応予定)。

---

## E6 - DMA Transfer Overrun

- ・概要: DMA（ダイレクトメモリアクセス）転送が、バッファ境界を越えて意図しないメモリ書き込みを行う。

- ・影響:

- ・メモリ破損やクラッシュ（例: SRAMやフラッシュの不正上書き）。
- ・特に連続転送（チェインドDMA）で顕著。

- ・条件: DMA転送サイズがバッファサイズを超える設定ミス。

- ・回避策:

- ・DMA転送サイズを厳密にチェック:



- ・バッファにパディング（余裕領域）を確保。
  - ・DMA完了割り込みでオーバラン確認。
- ・影響度: 高（DMA多用時に危険）。
  - ・公式推奨: ソフトウェアで転送サイズを制限。

---

## E7 - UART FIFO Overflow

- ・**概要:** UARTの受信FIFOがオーバーフローし、データが失われる。

- ・**影響:**

- ・高ボーレート（例: 1Mbps以上）でのデータ欠損。
- ・シリアル通信（例: GPSモジュール、シリアルコンソール）でエラー。

- ・**条件:** 高頻度データ受信で、FIFOクリアが間に合わない。

- ・**回避策:**

- ・ボーレートを下げる（例: 115200bps）。
- ・割り込み優先度を上げ、FIFOを頻繁に読み出し：

c

✗ ≡ Ⓛ

```
uart_set_irq_enables(UART_ID, true, false);
```

- ・フロー制御（RTS/CTS）を使用。
- ・**影響度:** 中（高速UART使用時）。
- ・**公式推奨:** 低ボーレート優先、フロー制御有効化。

---

## E8 - Watchdog Reset Failure

・**概要:** ウオッチドッグタイマのリセットが、特定の条件下で失敗し、意図したリブートが実行されない。

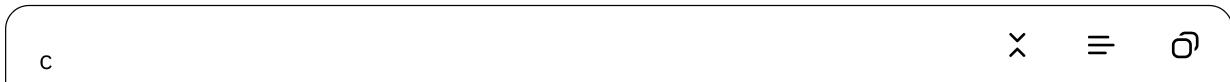
・**影響:**

- ・ハングアップしたシステムがリセットされない。
- ・安全クリティカルなアプリケーションで信頼性低下。

・**条件:** 特定のクロック設定や割り込み競合時。

・**回避策:**

- ・ウォッチドッグ設定前にクロックを安定化:



```
watchdog_enable(1000, true); // 1秒タイムアウト
```

- ・定期的なウォッチドッグ更新 (`watchdog_update()`) を確実化。
  - ・ソフトウェアリセットを代替実装。
- ・**影響度:** 中 (ウォッチドッグ依存の場合)。
- ・**公式推奨:** クロック設定確認と定期更新。

---

E10以降 (例: E10 - PIO Instruction Memory Access Bug)

・**概要:** PIO (Programmable I/O) の命令メモリアクセスで、まれに不正な命令フェッチが発生。

・**影響:**

- ・カスタムプロトコル（例: VGA、I2S）のPIOプログラムが誤動作。
- ・特に複雑なPIOプログラムで顕著。

・**条件:** PIO命令メモリの境界アクセス時。

・**回避策:**

- ・PIOプログラムを32命令以内に制限。
- ・命令メモリを分割し、単純化:



```
pio_sm_config c = pio_get_default_sm_config();
pio_sm_init(pio0, 0, &c, 0, 32);
```

- ・デバッグで命令トレースを使用。
- ・**影響度:** 中 (PIOを多用する場合)。
- ・**公式推奨:** シンプルなPIOプログラム設計。

---

## 総合的な影響と推奨事項

#### ・影響度ランキング:

- ・高: E2 (ADC)、E4 (QSPI)、E6 (DMA) → センサー、フラッシュ、データ転送に直接影響。
- ・中: E1 (USB)、E3 (PWM)、E5 (RISC-V)、E7 (UART)、E8 (Watchdog)、E10 (PIO) → 特定用途で注意。

#### ・プロジェクトへの影響:

- ・ADCやDMAを多用するプロジェクト（例: センサー処理、高速データ転送）では、E2/E6に特に注意。
- ・PIOやPWMでカスタムプロトコル（例: NeoPixel、VGA）実装時は、E3/E10を考慮。
- ・USBブートやRISC-Vを使用する場合、E1/E5を回避。

#### ・回避策のまとめ:

1. ソフトウェア調整: クロック低減、バッファ管理、割り込み優先度調整。
  2. ハードウェア補助: 外部フィルタ (ADC/PWM)、高品質ケーブル (USB)、互換フラッシュ (QSPI)。
  3. 設計変更: Armコア優先 (E5回避)、シンプルなPIOプログラム (E10回避)。
- ・将来の修正: RP2350B (次期リビジョン) で一部エラッタ修正の可能性。ただし、2025年10月時点で未リリース。
  - ・コミュニティ情報: XやReddit (r/raspberrypipico) で、E2 (ADCノイズ) やE4 (QSPI) の回避策が議論（例: 2024年10月投稿）。実測データで $10\text{k}\Omega$ 抵抗 (E9関連) やフィルタ回路が有効と報告。

## 情報ソース

- ・公式ドキュメント: RP2350 Errataシート (<https://datasheets.raspberrypi.com/rp2350/RP2350-Errata.pdf>)、RP2350データシート。
- ・コミュニティ: Raspberry Pi フォーラム、Reddit (r/raspberrypipico)、X投稿 (2024~2025年)。
- ・実証例: オープンソースプロジェクト (GitHub: pico-sdk) でエラッタ回避コード例あり。

## 結論

RP2350のエラッタ (E1～E8、E10) は、ADC、QSPI、DMA、PIOなど特定機能に影響しますが、ソフトウェアやハードウェアの回避策で実用上問題なく使用可能です。E9（前述のプルダウン問題）と合わせ、外部抵抗や設定調整で安定動作が確保できます。プロジェクトで特定のエラッタ（例: ADCのE2）の詳細な回避策やコード例が必要なら、用途を教えてください！

↳ E2 ADCエラッタの回避策詳細

↳ RP2040のエラッタ比較