

# レーダーベースボード MCU7

## XENSIV™ レーダーシステムプラットフォーム

ボードバージョン 2.2

### 本書について

#### 適用範囲と目的

このアプリケーションノートでは、インフィニオンの XENSIV™ 60 GHz レーダーシステムプラットフォームの一部であるレーダーベースボード MCU7 (DEMO RADAR MCU7) について説明します。また、複数のセンサーをサポートするプラットフォームの概要についても紹介します。ボードの中心部には、32 ビット Arm® Cortex®-M7 マイクロコントローラーユニット (MCU) である Microchip ATSAMS70Q21 が搭載されています。マイクロコントローラーは、最大 2 つのセンサーとのインターフェースを持ちます。また、可視化や高速データ処理のために、ホストコンピューターへの高速 USB 2.0 インターフェースを提供します。さらに、このボードは Arduino MKR 規格と互換性があり、大容量データストレージまたはワイヤレス通信 Arduino ボードなど既存の量販ドーターボードへのアクセスを容易にします。

#### 対象者

このドキュメントは、インフィニオンの XENSIV™ 60 GHz レーダーシステムプラットフォームを使用するすべての人を対象とします。

## 目次

本書について .....	1
目次 .....	2
1 はじめに .....	3
1.1 レーダーシステムプラットフォーム .....	3
1.2 ボードの概要と主な機能 .....	
2 センサーインターフェース .....	5
2.1 概要 .....	5
2.2 電源 .....	6
2.3 レベルシフター .....	7
2.4 センサーコネクタ .....	8
2.5 レーダーセンサー ボードの電流測定 .....	10
3 Arduino MKR インターフェース .....	12
4 ファームウェア .....	13
4.1 USB .....	14
4.2 EEPROM .....	14
4.3 RGB LED .....	15
4.4 ファームウェア開発およびデバッグ .....	16
4.5 ブートローダーを介したフラッシュ .....	16
5 マイクロコントローラーのピンマップ .....	17
5.1 センサーインターフェース 1 .....	17
5.2 センサーインターフェース 2 .....	18
5.3 Arduino MKR インターフェース .....	19
5.4 その他のピン .....	20
6 参考資料 .....	21
改訂履歴 .....	22
免責事項 .....	23

# 1 はじめに

## 1.1 レーダーシステムプラットフォーム

60 GHz レーダーシステムプラットフォームは、インフィニオンの 60 GHz レーダーセンサーの評価プラットフォームです。これは、マイクロコントローラーボードとしてのレーダーベースボード MCU7 と、BGT60TR13C シールド、BGT60LTR11AIP シールド、BGT60UTR11AIP シールド、または BGT60ATR24C シールドなどの 60 GHz レーダーセンサーボードで構成されます。Figure 1 に、レーダーベースボード MCU7 とセンサーボード (BGT60TR13C シールド) を示します。両方のボードにマーカーがあります。これらのマークは、センサーボードを正しく接続するために位置合わせする必要があります。

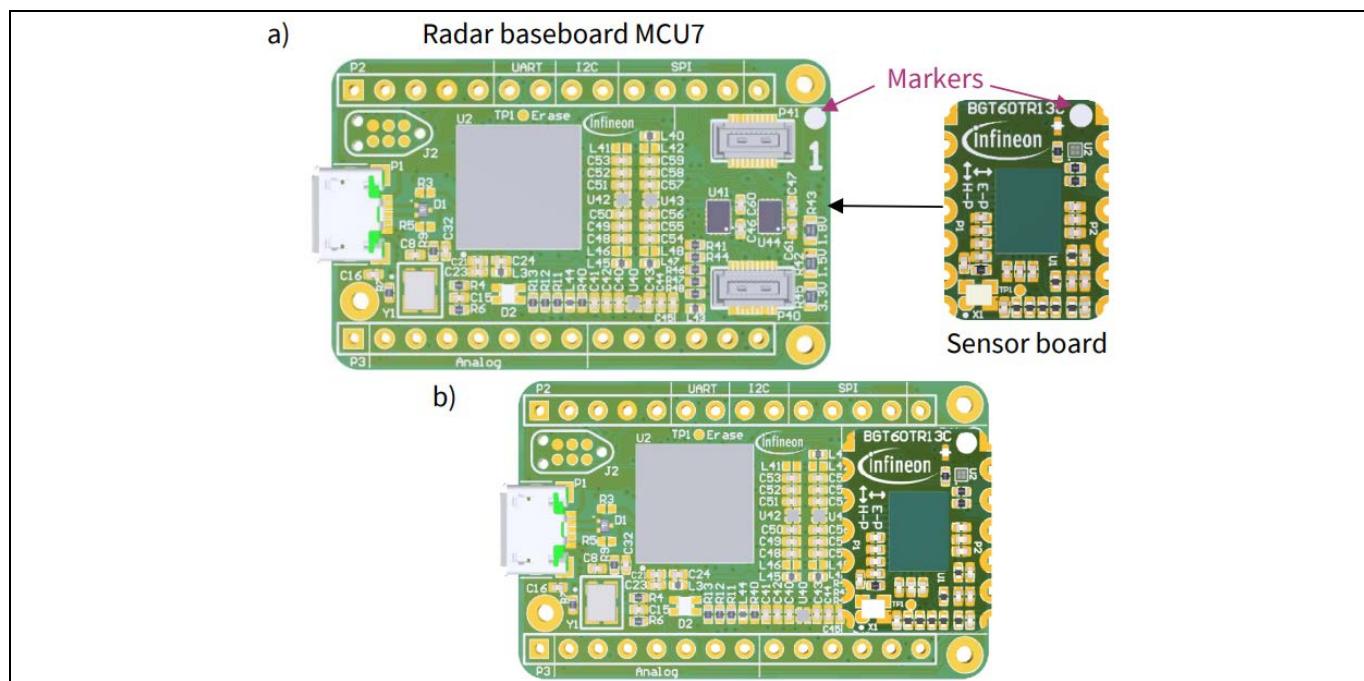


Figure 1 センサーボードの (a) プラグが抜かれた状態、または (b) プラグが差された状態の 60 GHz レーダーシステムプラットフォーム

## 1.2 ボードの概要と主な機能

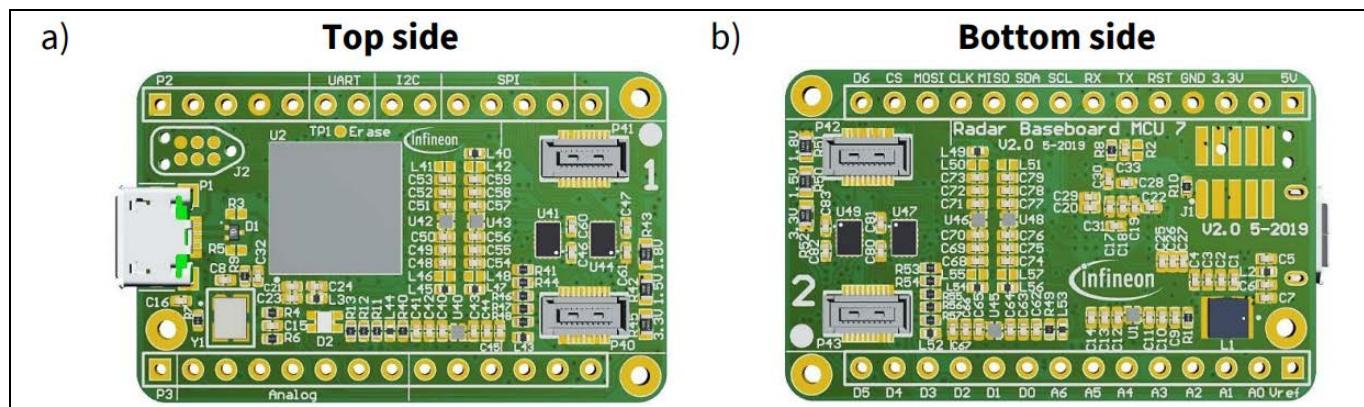


Figure 2 レーダーベースボード MCU7

Figure 2 に示すレーダーベースボード MCU7 は、40.64 mm x 25.4 mm のプリント回路基板 (PCB) です。その主な目的は、インフィニオンの 60GHz レーダーセンサーに汎用センサーインターフェースを提供することです。セントラル MCU は、レーダーのデータ処理を実行したり、センサーデータを USB インターフェースまたは Arduino MKR インターフェースに転送したりできます。ボードのブロック図を Figure 3 に示します。その主な機能は、次のとおりです。

- **Arm® Cortex®-M7 MCU** – レーダーベースボード MCU7 の主なデバイスは、32 ビット Arm® Cortex®-M7 MCU である ATSAMS70Q21 です。300 MHz のクロック周波数で動作し、384 キロバイトの SRAM が搭載されています。マイクロコントローラーには 3.3 V が供給され、デジタル信号とセンサー電源間のクロストークを最小限に抑えるために、センサー用の他の 3.3 V 電源から切り離されています。
- **USB インターフェース** – レーダーデータ取得には大量の生データが生成される可能性があるため、レーダーベースボード MCU7 では最大 480 Mbit/s のデータレートの高速 USB インターフェースが使用されます。これにより、すべてのレーダーデータを可視化するためのホスト PC、またはレーダーデータ処理のために他の強力なコンピューターへの転送に十分な帯域幅が提供されます。
- **センサーインターフェース** – レーダーベースボード MCU7 は、2 つのレーダーセンサーインターフェースを備えています。各センサードーターカードには、I<sup>2</sup>C インターフェースを介して接続された EEPROM が含まれます。この EEPROM には、ベースボードとセンサーのタイプを示す識別情報が含まれています。これは、ファームウェアがボードと適切に通信するために使用されます。ボードが誤った方向への挿入(180 度回転)されたことも検出可能です。レーダーセンサーとの通信を可能にするために、ボードは SPI 接続端子、2 つのアナログ-デジタル変換 (ADC) チャネル、1 つのデジタル-アナログ変換 (DAC) チャネル、および 10 個の GPIO ピンを提供します。
- **Arduino MKR コネクタ** – レーダーデータ処理には異なるアプリケーションに特定的なハードウェアが必要になる場合があるため、レーダーベースボード MCU7 は Arduino MKR インターフェースをサポートします。これにより、生データ格納用の SD カード、ワイヤレス通信など、ラピッドプロトタイピング用のさまざまな安価なアドオンハードウェアにアクセスできます。
- **RGB LED** – ボードにはステータスを示す RGB LED があります。ボードの起動時に、LED が点滅します。レーダーセンサーが検出されると、LED が緑色に変わります。LED ステータスコードの詳細な説明はセクション 4.3 にあります。

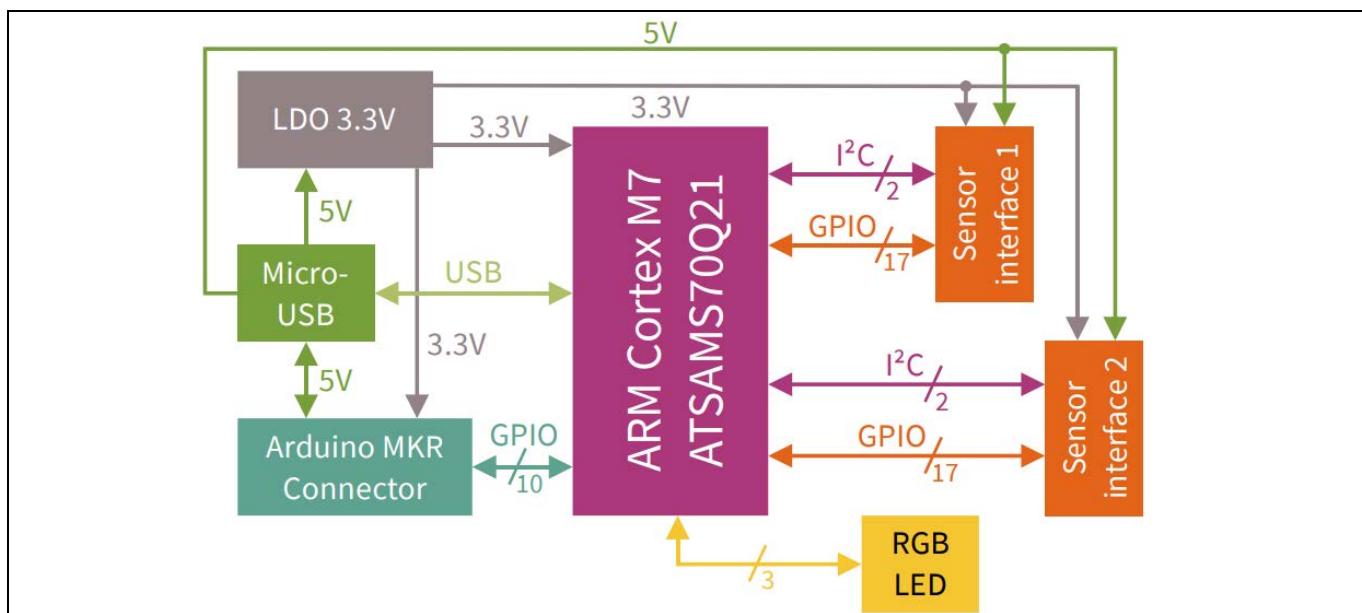


Figure 3 レーダーベースボード MCU7 のブロック図

## 2 センサーインターフェース

ここでは、前のセクションで紹介したセンサーインターフェースについて説明します。

### 2.1 概要

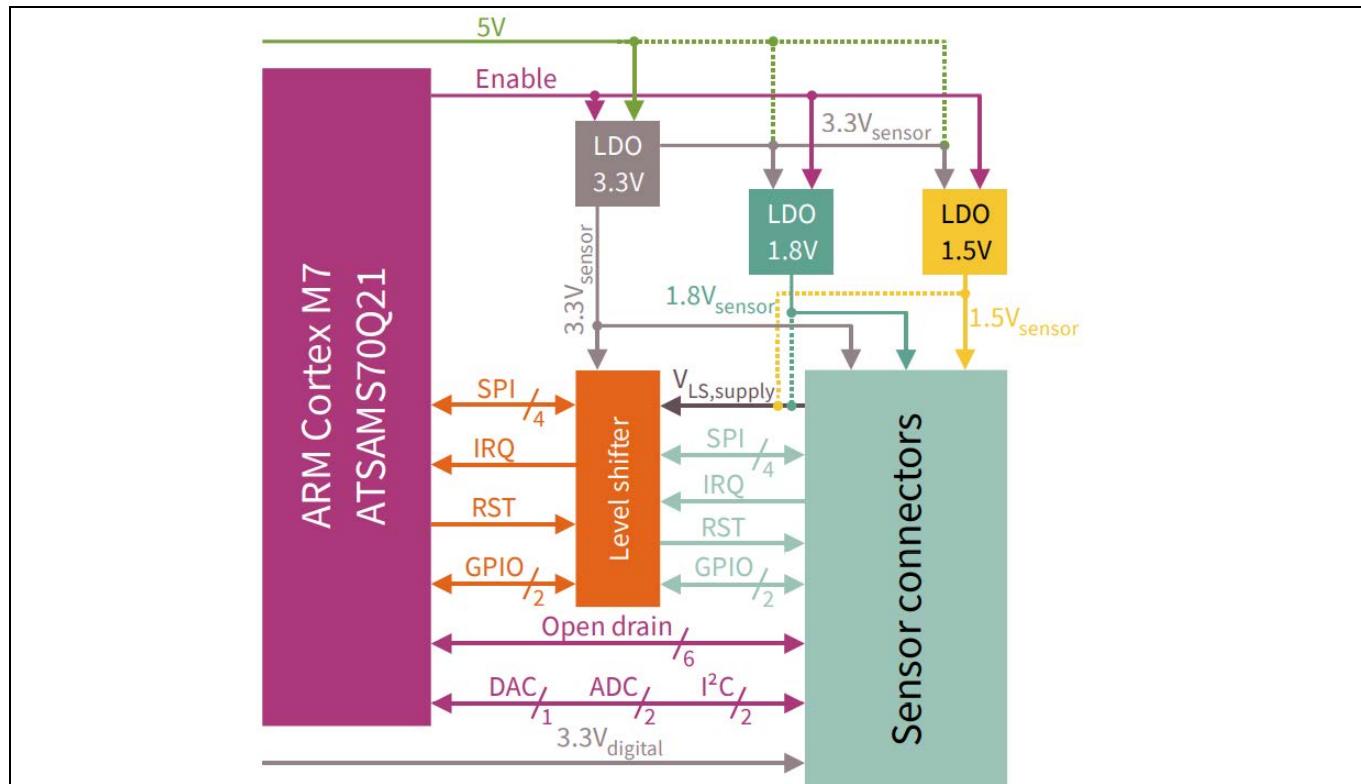


Figure 4 レーダーベースボード MCU7 のセンサーインターフェースのブロック図

レーダーベースボード MCU7 の単一センサーインターフェースのブロック図を Figure 4 に示します。ボードが起動したとき、またはセンサーがセンサーコネクタに接続されたとき、センサーの電源は無効になります。起動中は、センサー上で EEPROM の 3.3 V<sub>digital</sub> (MCU) 電源のみがアクティブになります。MCU は、センサー上でコネクタに接続されているかどうかを検出します。センサー上で誤った向きに接続されると、MCU はそれを検出し、RGB LED でエラーを通知します。詳細については、セクション 4.2 を参照してください。センサーが正しく接続されている場合、MCU は EEPROM 内の情報を読み出します。これは、どの種類のセンサーがインターフェースに接続されているかに関する情報を取得する方法です。センサーの電源は、正しいセンサーが検出された場合にのみ有効です。

センサーとの主な通信インターフェースはシリアルペリフェラルインターフェース (SPI) です。3.3 V の MCU 供給電圧と 1.8 V または 1.5 V のセンサー供給電圧の間の不一致を考慮して、レベルシフターによって異なる電圧レベル間を変換しています。レベルシフターの供給電圧は片側で 3.3 V で、もう一方では、3 つのソースから給電されます。1.5 V または 1.8 V 電源から直接、またはセンサーコネクタから、電力が供給されます。センサーコネクタによって提供される場合、データカードは供給された電圧から正しいレベルシフター供給電圧を提供できます。4 つの SPI 信号に加えて、さらに 4 つのデジタル信号が変換されます。

- センサーへのハードウェアリセット信号
- MCU による何らかのアクションが必要であることを示す、センサーから MCU への割込み信号
- 2 つの汎用入力出力 (GPIO) ピン - 詳細についてはセクション 2.3 を参照

変換された信号に加えて、コネクタが以下も提供します。

- 2つのADC チャンネル
- 1つのDAC チャンネル
- 6つのオープンドレイン出力信号。これらの1つは、センサー ボード上のLEDを制御することを目的としています。MCUは、信号をローレベルにプルすることによりLEDをアクティブにし、信号をHigh-ZモードにすることによってLEDを非アクティブにします。

## 2.2 電源

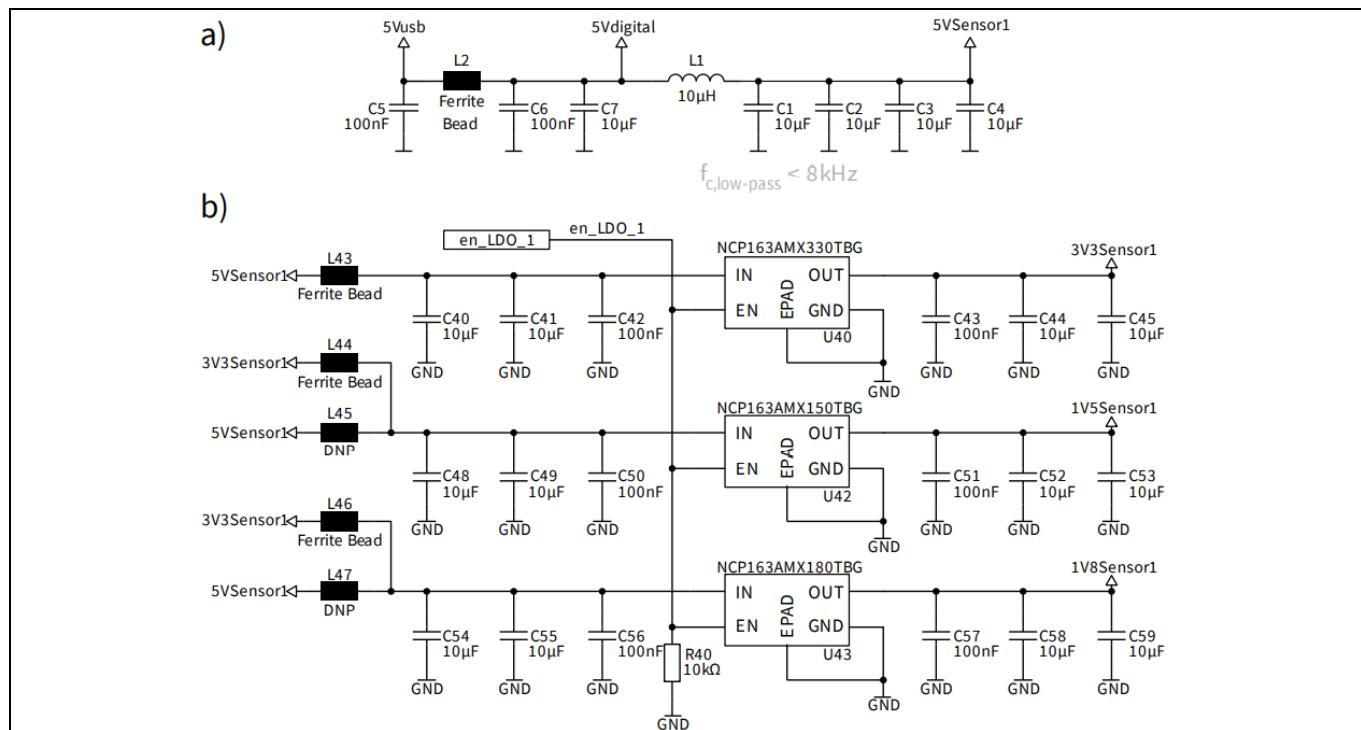


Figure 5 センサー1のセンサー供給回路の回路図

レーダーセンサーは電源の電圧変動に非常に敏感であるため、低ノイズの電源が重要です。ここでは、レーダーベースボード MCU7 の電源回路について説明します。Figure 5a に、電源回路の初期段階を示します。USB 電源を経由して回路に入る高周波 (HF) ノイズをブロックするために、ローパスフィルタが内蔵されています。このフィルタは、フェライトビーズ L2 とコンデンサ C6 および C7 で構成されています。フィルタリングされた電圧は、MCU に供給するために使用されます (3.3V LDO を介して)。MCU の動作により、電源に不要な電圧変動が発生する可能性があります<sup>1</sup>。したがって、別のローパスフィルタが電源に配置されます。フィルタはコイル L1 とコンデンサ C1、C2、C3、C4 で構成されており、カットオフ周波数は 8 kHz 未満になります<sup>2</sup>。これにより、約 10kHz から MHz の周波数範囲で非常に低いノイズで安定した電源電圧が得られます。

Figure 5b に、各センサーに必要な追加の電源回路を示します。Figure 5b に示されるのはセンサー1用です<sup>3</sup>。フェライトビーズが電源の入力側に配置され、上流からの HF ノイズ (MCU によって引き起こされる可能性があるなど) をブロックします。これは LDO に供給され、センサー電源用に 3.3V の中間電圧を生成します。この中間電圧はさらに 1.8V と 1.5V にダウンコンバートされます。電圧変換に使用される

<sup>1</sup> 例えば、MCU によって生成された PWM 信号は、PWM 信号とその高調波の周波数で電源にノイズをもたらします。

<sup>2</sup> 供給ラインにコンデンサを追加すると、カットオフ周波数がさらに低下します。

<sup>3</sup> センサー2の回路は、センサー1の回路のコピーです。

LDO IC は、オン・セミコンダクター製出力ノイズ  $6.5 \mu\text{V}_{\text{RMS}}$  の低ノイズ LDO NCP163 です。これらは、90 dB を超える電源除去比を提供します。システム内の残りのノイズは主に LDO によって引き起こされ、LDO が効率的に機能しなくなる 100 kHz 付近の周波数で最も強くなる傾向があります。

レーダー信号にゴーストターゲットが現れる場合は、電源を最適化することで性能を向上させることができます。レーダーベースボード MCU7 では、L44 を L45 に、L46 を L47 に交換することにより、1.8 V および 1.5 V の LDO の電源を 3.3 V から 5 V に変更できます。これは、1.5 V 電源と 1.8 V 電源のノイズ性能に影響します。L44 と L46 を使用する場合、1.5 V と 1.8 V はそれぞれ 2 つの LDO を直列に接続して生成されます<sup>1</sup>。この設定は、単一の LDO よりも外部低周波数を抑制します。

もう 1 つのオプションは、L44 と L46 の代わりに L45 と L47 を使用して、5 V を 1.5 V と 1.8 V に直接変換することです。各 LDO は、最大抑制周波数(通常は約 100 kHz)でノイズを生成します。この周波数は、レーダーIF 信号の周波数範囲内にあります。したがって、この周波数範囲のノイズは、レーダーデータ処理でゴーストターゲットを引き起こす可能性があります。2 つの LDO が直列に接続された構成(L44 と L46)では、最初の LDO ステージによって発生したノイズは 2 番目のステージによって増幅されます。100 kHz 付近のノイズを最小限に抑えるために、L45 および L47 の構成で単一の LDO ステージを使用できます。まとめると、強い低周波ノイズ成分を備えた電源の場合は L44 と L46 を使用する方が適切ですが、低ノイズ電源の場合は L45 と L47 を使用することを推奨します。

## 2.3 レベルシフター

ここでは、レーダーベースボード MCU7 のレベルシフター回路について説明します。

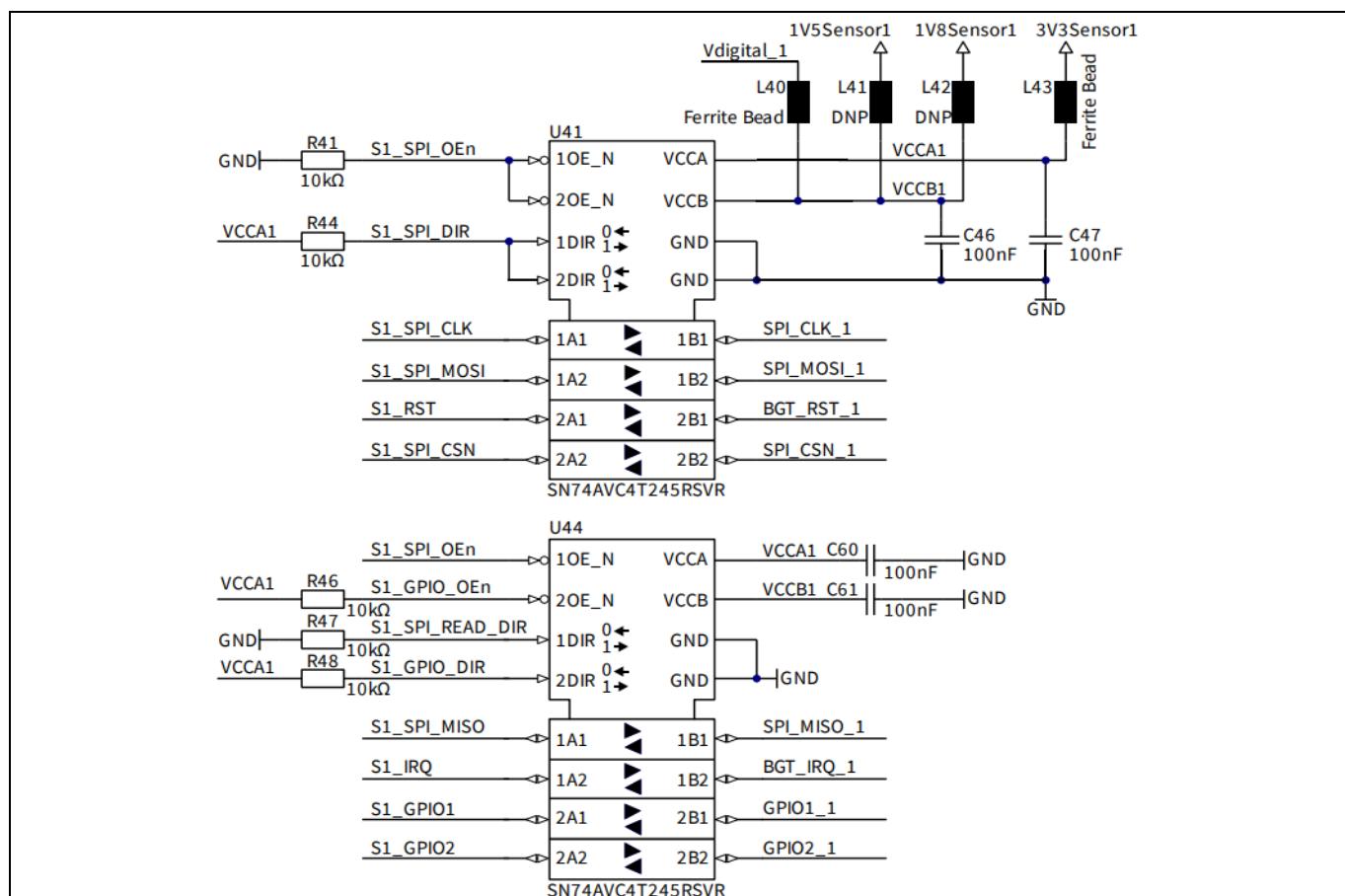


Figure 6 センサー1のレベルシフター回路

<sup>1</sup> U40 と U42 は 1.5 V を生成します。U40 と U43 は 1.8 V ドメインを作成します。

Figure 6 に、センサー1 のレベルシフター回路を示します。センサー2 の回路は、センサー2 に使用される他の信号線を除いて、センサー1 の回路と同じです。レベルシフターは、センサー電源の 1 つから、またはセンサーインターフェースから電力を供給できます。センサーインターフェースからのこの電圧により、ドーターカードはその優先供給電圧を選択するオプションがあります。電源電圧のデジタル信号からの干渉を最小限に抑えるために、レベルシフターはセンサー電源から切り離されています。これは、MHz 領域で高い抑制性を備えたフェライトビーズを使用することによって行われます。

プルアップ抵抗とプルダウン抵抗は、イネーブルピンと方向ピンに接続されています。これにより、MCU がマスターで、センサーがスレーブになるように構成された動きで SPI レベル変換が有効になるようにプリセットが提供されます。別の設定が必要な場合、MCU はそれらのピンを直接駆動できます。詳細については、Table 5 および Table 6 を参照してください。

## 2.4 センサーヨネクタ

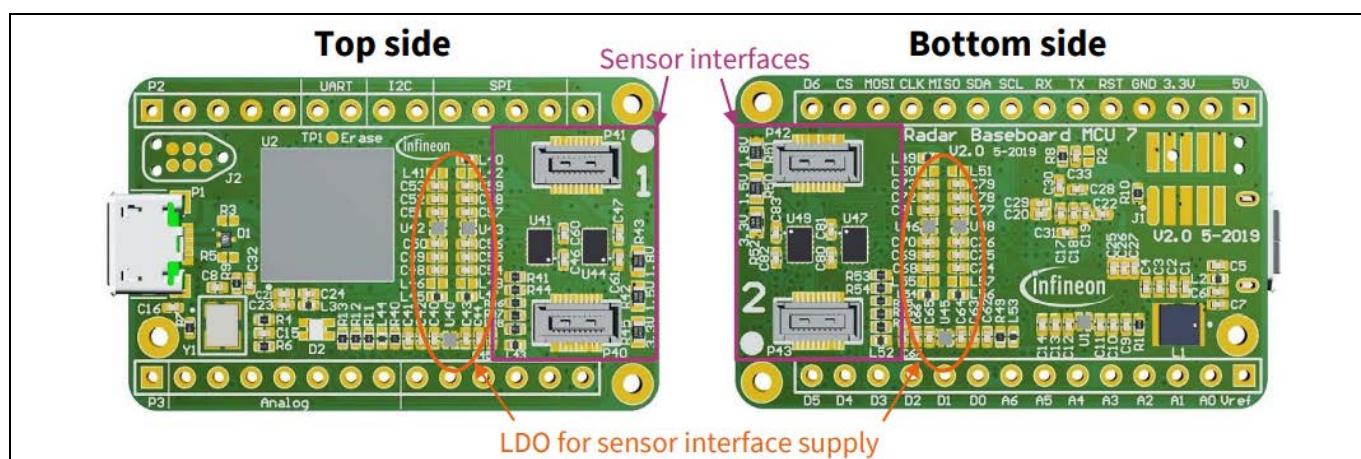


Figure 7 レーダーベースボード MCU7 のセンサーインターフェース

Figure 7 に示すように、ボードは 2 つのセンサーインターフェースを備えています。PCB の上面にあるものには「1」とラベルされ、PCB の下面にあるものには「2」とラベルされています。ラベルの横に、白く塗りつぶされた円がマーカーとして描かれています。補完的なマーカーがセンサー ボードにあります。Figure 1 を参照してください。センサーを正しく取り付けるには、レーダーベース ボード MCU7 とセンサー ボードのマーカーを揃える必要があります。

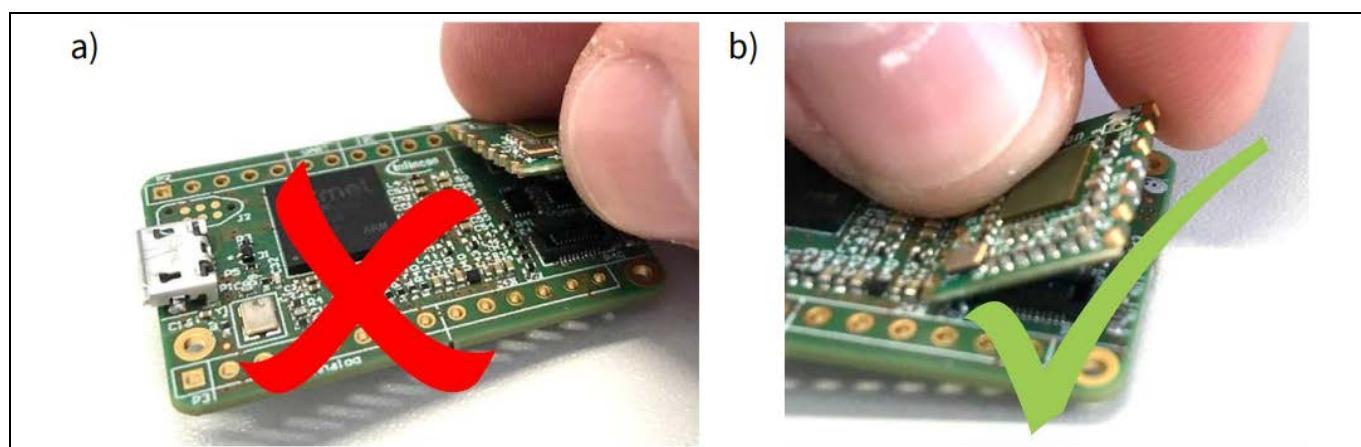


Figure 8 レーダーベースボード MCU7 のセンサー間コネクタを抜く方法

レーダーベースボード MCU7 のセンサーインターフェースには、ヒロセの DF40C-20DS-0.4V コネクタが使用されています。センサー ボードには、DF40C-20DP-0.4V コネクタと一致するものが使用されます。頻繁にセンサーに抜き差しするとコネクタが摩耗する恐れがあります。これを防ぐためには、Figure 8a に示すように、センサーをコネクタの短辺から持ち上げないでください。Figure 8b に示すように、代わりにセンサーの長辺を引くことで短辺を傾けられます。これにより、コネクタの寿命が大幅に延びます。

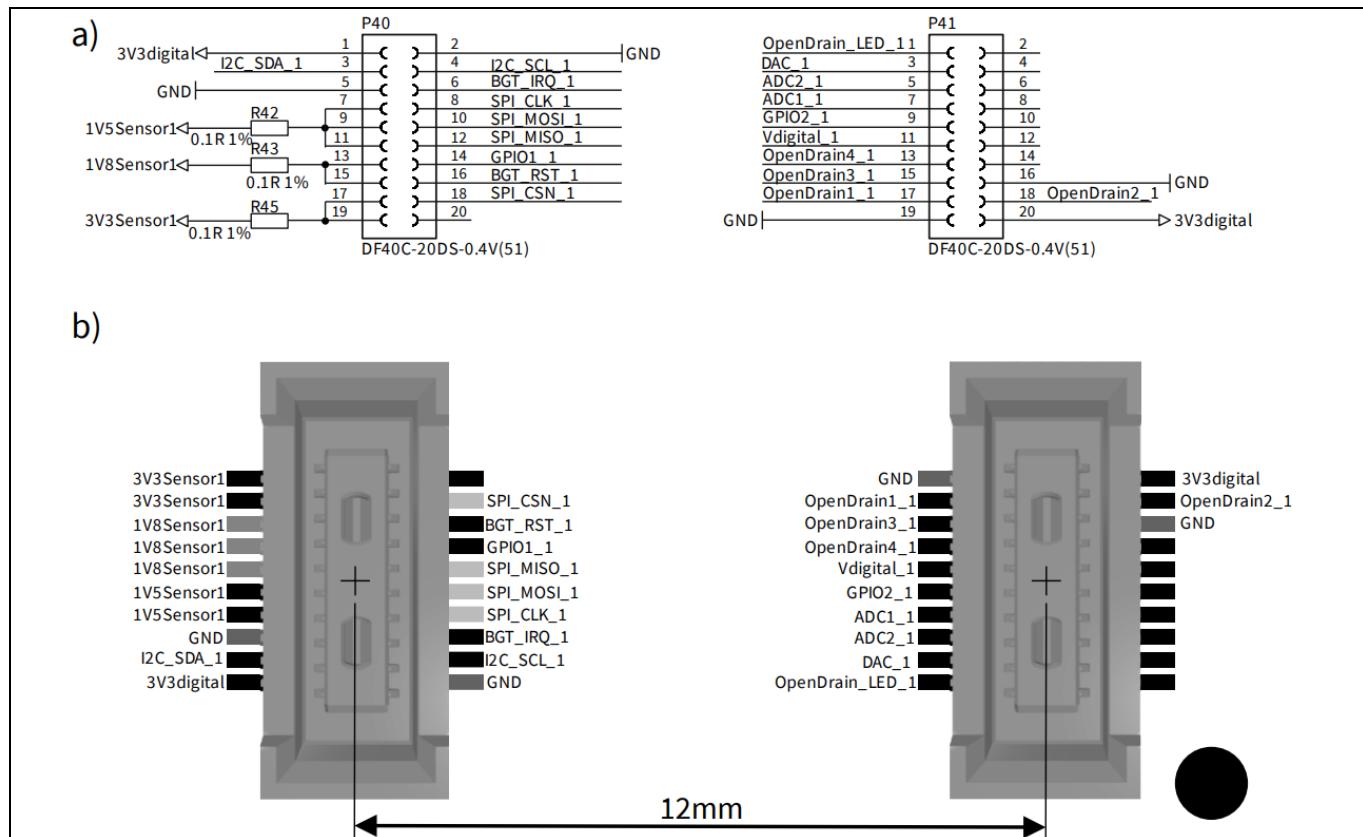


Figure 9 センサー1 のコネクタの回路図 (a) とパッドレイアウト (b)

Figure 9 に、センサー1 のコネクタの回路図 (a) とパッドレイアウト (b) を示します。同じセンサーをどちらのインターフェースにも接続できるため、センサー2 の信号名が異なることを除いて、両方の回路図とパッドレイアウトは同じです。センサーインターフェースの 2 つのコネクタは 12mm 離れており、Figure 9 に、センサーをレーダーベースボード MCU7 のセンサーインターフェースの 1 つと正しく位置合わせするためのシルクスクリーンマーキングも示します。

センサーが正しく接続されていない場合 (180 度回転)、GND と 3V3digital は MCU ボードとセンサー ボードの間で一致します。したがって、EEPROM 用のセンサーの I<sup>2</sup>C 信号の代わりに OpenDrain1\_1 と OpenDrain2\_1 がプルアップされていることを認識します。これにより、MCU は、センサーが正しく接続されている (I<sup>2</sup>C 信号がプルアップされている) か、正しく接続されていないか (OpenDrain1\_1 および OpenDrain2\_2 がプルアップされている) を検出できます。

メインセンサーインターフェースは全二重 SPI です。セクション 2.3 で説明した 8 つのレベルシフト信号に加えて、1 つのラインがレベルシフターへ電源を提供し、ドーターカードがデジタル電圧レベルを設定できるようにします。加えて、これらの信号には、センサー ボードで使用できる MCU に接続された 3 つのアナログライン、2 つの ADC チャンネル、および 1 つの DAC チャンネルがあります。さらに、5 つのデジタル信号が接続されていますが、レベルシフトされていないため、オープンドレインモードでのみ使用できます。

## 2.5 レーダーセンサーの電流測定

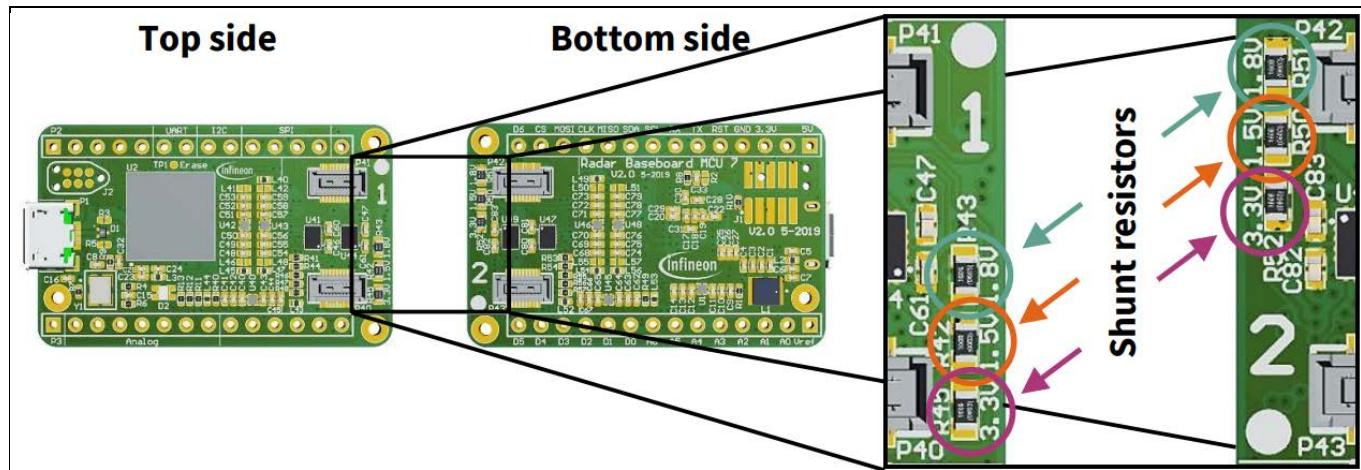


Figure 10 レーダーセンサーの電流測定用のレーダーベースボード MCU7 のシャント抵抗

レーダーセンサーをモバイルアプリケーションで使用する場合、バッテリーの寿命を延ばすために消費電力を最小限に抑える必要があります。レーダーグリフは、構成された設定の予想平均消費電力の見積もりを提供します。センサーの実際の動作電流消費量の測定が必要な場合は、レーダーベースボード MCU7 にシャント抵抗が用意されています。Figure 10 に示すように、シャント抵抗は 2 つのセンサーの電源に直列に接続されています。各シャント抵抗の横には、測定される電源を示すラベルがあります。詳細については、Table 1 を参照してください。すべてのシャント抵抗の抵抗値は  $0.1\Omega$  です。シャント抵抗に沿った電圧降下を測定することにより<sup>1</sup>、ユーザーはセンサーに供給される電流を推測し、供給される電圧をかけてセンサーの消費電力を測定できます。このようにして、センサーの 1.5 V, 1.8 V, および 3.3 V 電源に供給される電力を測定できます。

Table 1 シャント抵抗とそれらが接続される電源

シャント抵抗	抵抗値	センサーインターフェース	電源
R42	$0.1\Omega$ (1%)	1	1.5 V
R43	$0.1\Omega$ (1%)	1	1.8 V
R45	$0.1\Omega$ (1%)	1	3.3 V
R50	$0.1\Omega$ (1%)	2	1.5 V
R51	$0.1\Omega$ (1%)	2	1.8 V
R52	$0.1\Omega$ (1%)	2	3.3 V

Figure 11a に示すように、オシロスコープで電流を測定する場合、2 つのオシロスコープチャネルが必要です。このような設定は、共有グラウンドを使用した差動測定に必要です。対象となる電圧は、チャネル 1 の電圧からチャネル 2 の電圧を引いたものです。Figure 11b は、レーダーフレームのチャーブ中の典型的な測定結果を示しています。チップの RF 部分がアクティブになります。PLL が周波数をロックし始めると、BGT60TR13C への電流が急増します。オーバーシュート後、パワーアンプと ADC がアクティブになります。周波数ランプの後、必要な電流が低下し、電流のアンダーシュートが観察されます。この結果、バッファコンデンサ間の電荷が振動するため、レーダーセンサーを明確な状態にして、センサーの電圧を安定させることを推奨します。

<sup>1</sup> シャント抵抗に沿った電圧降下を差動またはポテンショナルフリーで測定してください。

結果として生じる定電流は、測定された電圧をシャント抵抗で除算することにより、オシロスコープまたは電圧計で簡単に測定できます。消費電力を計算するには、レーダーフレームのすべての状態の電流(この時の電力)を測定する必要があります。次の計算ステップは、電力に持続時間を掛けて、各状態で消費されるエネルギーを計算することです。次に、フレーム全体のエネルギーを取得するために、フレームのすべてのエネルギーを合計する必要があります。平均消費電力を得るには、フレームのエネルギーをフレームの長さで割る必要があります。

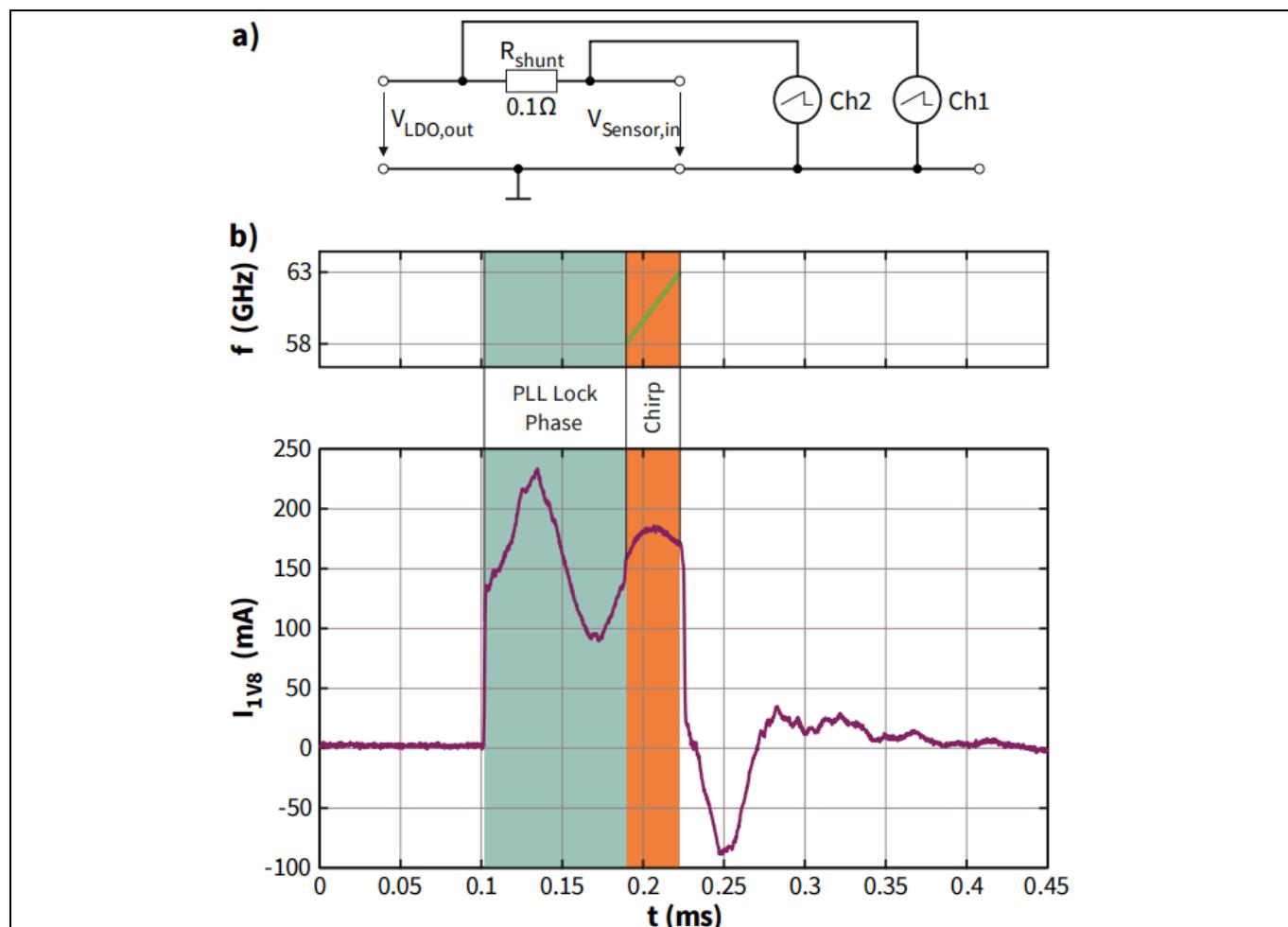
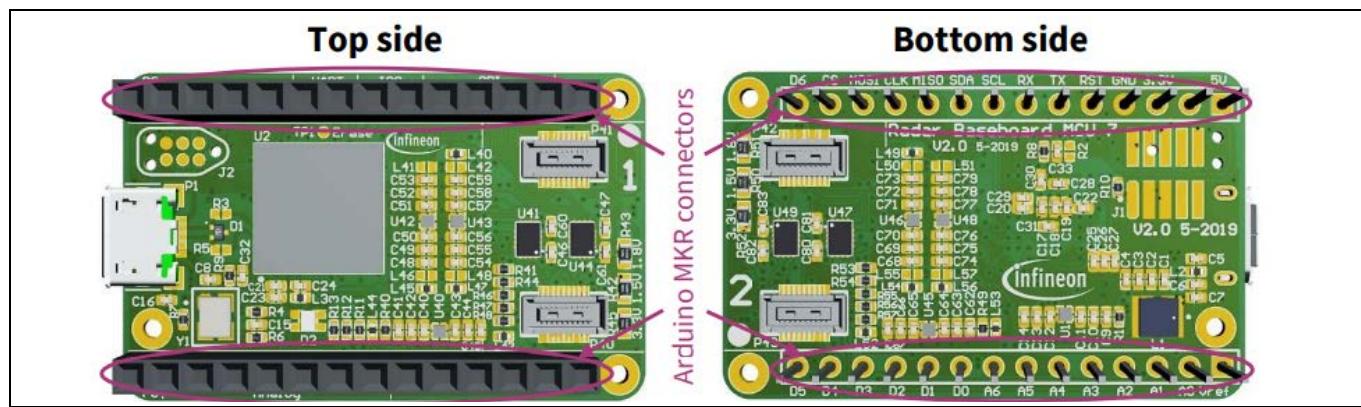


Figure 11 シャント抵抗を使用した電流測定。測定のセットアップを (a) に示し、通常の単一チャネルの測定結果を (b) に示します。

### 3 Arduino MKR インターフェース

Arduino MKR ファミリは、ラピッドプロトタイピングに使用できる標準化されたインターフェースを備えた新しい Arduino ファミリのボードです。この新しいボードファミリには、(とりわけ)次のボードが含まれています。これらのボードはドーターカードとして使用できます。

- MKR Zero – SD カードに生データを保存できる SD カードスロットを備えたボード
- MKR1000 – ワイヤレスデータ転送用の Wi-Fi モジュールを備えたボード
- MKR Vidor 4000 – 並列データ処理用の FPGA と RAM 拡張用の SDRAM を備えたボード
- MKR2UNO アダプター – 膨大な数のドーターカードをアクセスできる標準の Arduino 環境を提供するアダプターボード
- その他; 「[Arduino MKR ボードの概要](#)」をグーグルで検索することで簡単に概要がわかります。



## 4 ファームウェア

レーダーベースボード MCU7 には、ホスト (通常は PC) とセンサーコネクタに取り付けられたセンサーシールドの間のブリッジとして機能することを目的としたデフォルトのファームウェアが付属します。このために、ファームウェアは次のロジックを実装します。

- USB 経由でホストとの通信
- センサーレジスタの読み出しと書き込み
- SPI を介したセンサーデータの読み出し
- センサー ボードがコネクタの 1 つに接続されているかどうかの確認
- センサー ボード上の EEPROM の読み出しと書き込み (例えば、ボードを識別するため)
- ベース ボードやセンサー ボードのステータス LED などの補助周辺機器の制御

現在の状態では、ファームウェアは信号処理を実装していません。ホストからセンサーの構成内容を伝えるメッセージを受信し、取得したデータをセンサーからホストに転送できます。ただし、信号処理が必要な場合は、MCU が十分に強力であるため、この機能を追加できます。ファームウェアは、他のマイクロコントローラーに移植する手間を最小限に抑えるように作成されています。したがって、ファームウェアは次の層で構成されています。

- **ハードウェアアブストラクションレイヤー (HAL)** – これは最下位レイヤーであり、SPI, GPIO, タイマーおよびその他の HW ブロックとのインターフェースで連結する方法など、MCU 固有のすべてのロジックが実装されています。別の MCU に移植する場合、影響を受けるのはこのレイヤーだけです。また、通信レイヤーがホストデバイスに接続するために使用する USB CDC シリアルインターフェースも実装しています。
- **ドライバーレイヤー** – このレイヤーは、ハードウェアレイヤーの上にあります。センサーからデータを読み取る機能とパラメータ化する機能を提供します。適切な SPI パケットを作成してこれらのパラメーターをセンサーに適用し、HAL から SPI 関数を呼び出してデータ転送を開始します。さらに、HAL が受信したセンサーからの SPI データパケットをデコードします。これにより、センサーの測定値が抽出され、上のレイヤーに渡されます。ドライバーレイヤーは MCU から独立しています。ただし、これはセンサーの固有仕様であり、サポートされているすべてのセンサーに適合させる必要があります。
- **通信レイヤー** – この層は、ドライバーレイヤーの上にあります。USB インターフェースを介してホストからの要求またはコマンドメッセージを受信します。これらのメッセージはデコードされ、含まれているデータが抽出されます。これは、ドライバーレイヤーの対応する関数を呼び出すことによって行われます。これら機能が結果を返すと、この結果を含む応答メッセージがホストに転送されます。ホストへのメッセージは、ドライバーレイヤーがセンサーデータを受信したときにも生成されます。このレイヤーもハードウェアに依存しません。ホストへのデータ送信に HAL USB CDC インターフェースを使用します。
- **実際のファームウェア** – これは他のすべてのレイヤーの上にあります。ファームウェアは、起動時にすべての下位層の初期化関数を呼び出して、SPI 速度や GPIO 方向などの設定を立ち上げます。起動フェーズが完了すると、無限ループに入ります。このループ内で通信層が呼び出され、ホストとの間でメッセージを送受信します。

レーダーベースボード MCU7 を仲介として使用せずに、カスタムプロジェクトにセンサーを統合する場合は、ファームウェアで次の手順を実行する必要があります。

- ハードウェアアブストラクションレイヤーは、プロジェクトで使用される MCU またはアプリケーションプロセッサに移植する必要があります。
- ドライバーレイヤーは、変更なしでそのまま使用できます。
- 通信層は必要ありません。代わりに、アルゴリズムをこのレベルに置くことができます。

- レーダーベースボード MCU7 のファームウェアコードは使用されません。ただし、初期化コードはプロジェクトの MCU に移動する必要があります。

ファームウェア全体を別の MCU に移植する場合は、HAL をその MCU に対応させるだけで十分です。

ファームウェアにアルゴリズムを追加するときは、これをドライバーレイヤーの上に配置する必要があります。要件に応じて、このアルゴリズムは通信レイヤーを置き換えることや、通信レイヤーと並行して操作もできます。

## 4.1 USB

ファームウェアは、USB インターフェースに通信デバイスクラス (CDC) デバイスを実装しています。つまり、USB インターフェースは、双方向にバイトストリームを送信するシリアルポートのように機能します。

このシリアルデータストリームに、メッセージベースのプロトコルは、各メッセージに開始マーカーと終了マーカーを配置しています。メッセージの最初のバイトは、メッセージの対象となるファームウェアのターゲットモジュールを指定します。メッセージのバイト 2 と 3 は、メッセージの長さを設定します。このようにして、プロトコルデコーダはメッセージの長さを認識します。開始マーカーと終了マーカーを組み合わせて、受信したストリームを構成したパケットが分割できます。一部の通信ライブラリを含むファームウェアのソースコードは、ホストアプリケーションに統合され、ファームウェアメッセージを作成およびデコードする機能を提供します。

## 4.2 EEPROM

ファームウェアは EEPROM を利用してシールドタイプを検出します。EEPROM の構造を Table 2 に示します。また、同じタイプの異なるボードを区別するために 128 ビットのボード識別子も含まれています。それに加えて、ファームウェアは、限られたサイズのキー/値ペアを EEPROM に保存および取得するためのコマンドをホストに提供します。ホストは、それらのペアに必要な意味を自由に割り当てることができます (例えば、特定のキーの値がレーダー構成データを参照し、別のキーの値が表示設定の保存に使用される場合があります)。EEPROM にアクセスするための機能は、ファームウェアに付属されている通信ライブラリの一部です。

Table 2 RGB LED によるシステムステータス表示

構造	フィールド	説明
EEPROM ヘッダー	ボードと EEPROM を識別するためのヘッダー	
	マジック	EEPROM の初期化を検出するために使用される 8 ビット識別子
	バージョン	構造を識別する 2 バイトのバージョン番号
	BoardType	ファームウェアがボード種類を検出するために使用するボードタイプ識別子
	NvmType	EEPROM タイプの識別子。ファームウェアが異なる I <sup>2</sup> C EEPROM 構成 (サイズ、ページサイズ) を区別するために使用します。
	予備	予備スペース、0 に設定
	UUID	ボードの識別するための 128 ビット UUID。
ディレクトリ	フラッシュ内のオフセット仕様で指定された 32 ビットキーの表マッピングする	
	エントリ 1: キー	最初のエントリ用の 32 ビットキー
	エントリ 1: 長さ	最初のエントリのデータ長

構造	フィールド	説明
	エントリ 1: ページ	ページ番号として指定された最初エントリのデータのアドレス
	...	その他のエントリ
	エントリ N: キー	N 番目のエントリの 32 ビットキー
	エントリ N: 長さ	N 番目のエントリのデータ長
	エントリ N: ページ	ページ番号として指定された N 番目のエントリのデータのアドレス
データ領域	EEPROM の残りの部分はエントリに対応するデータを格納するために使用されます。	
	データ 1	該当するディレクトリエントリで指定されたページ上エントリ 1 のデータ
	...	さらなるエントリのデータ
	データ N	該当するディレクトリエントリで指定されたページ上エントリ N のデータ

#### 4.3 RGB LED

ボードの現在のステータスは、RGB LED を使用して示されます。ボードの起動後、ボードがアイドル状態のとき、ボードのステータスは Table 3 に示すように表示されます。アクティブな場合、ボードのステータスは Table 4 に示すように表示されます。

Table 3 RGB LED で示される、起動後およびアイドル時のボードステータス

ボードステータス	点滅パターン			
	期間 1 と色 1	期間 2 と色 2	期間 3 と色 3	期間 4 と色 4
エラー未検出 (正常)	ロング緑	ロングオフ		
RF シールド未検出	ショート緑	ショート赤		
シールドの向きが間違っている	ショート赤	ショートオフ		
不明なシールド (シールドが壊れているか、正しく接続されていない可能性があります)	ショート黄	ショート赤		
初期化/検出中の内部エラー (ハードウェアまたはファームウェアの問題を示している可能性があります)	ショート黄	ショートオフ	ロング赤	ロングオフ

Table 4 RGB LED で示されるアクティブ中のボードステータス

システムステータス	点滅パターン			
	期間 1 と色 1	期間 2 と色 2	期間 3 と色 3	期間 4 と色 4
ボードがアイドル状態	ロング緑	ロングオフ		
SPI からデータの読み出し	赤			
ホストへデータの送信	青			

#### 4.4 ファームウェア開発およびデバッグ

ファームウェアは、Atmel Studio 7 IDE のプロジェクトとして提供されます。したがって、コンパイル、フラッシュ、およびデバッグは、IDE の対応するボタンを押すだけで、すぐに使用できます。詳細については、(Microchip, Atmel Studio 7 User Guide, 2019) を参照してください。

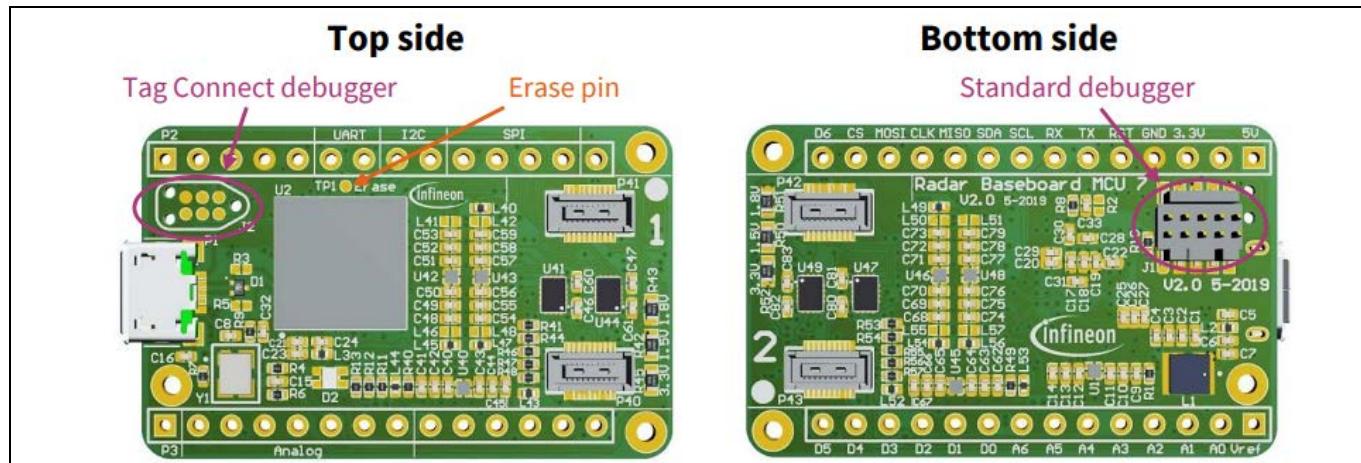


Figure 13 レーダーベースボード MCU7 のデバッグコネクタ

ユーザーがデバッガーコネクタを必要とする場合、ボードには、Figure 13 に示すように 2 つのオプションがあります。レーダーベースボード MCU7 の上面には、タグコネクタデバッガーのレイアウトがあります。ユーザーがタグコネクタデバッガケーブルを購入したくない場合は、通常の 10 ピン Arm® デバッガコネクタをレーダーベースボード MCU7 の底面にあるコンポーネント J1 のパッドにはんだ付けできます。

デバッガーコネクタを使用する場合、ファームウェアは Atmel-ICE プログラマーまたは互換性あるツールを使用して Atmel Studio 7 から直接プログラミングできます (詳細については、Atmel Studio の付属ドキュメントを参照してください)。

#### 4.5 ブートローダーを介したフラッシュ

マイクロコントローラーにはブートローダーが含まれています。これにより、デバッガーが使用できない場合に USB インターフェースを介したファームウェアの更新が可能になります。デバイスは通常、ブートローダーに入らずにファームウェアを直接起動するため、最初にブートローダーを手動でアクティベーションする必要があります。これは、次のシーケンスで実行できます。

- デバイスを USB から切断。
- TP1 を 3.3 V に接続して、MCU ERASE ピンを High に保持 (Figure 13 を参照)。
- TP1 を High に保ちながら、デバイスを USB に再度接続。
- ERASE ピン (TP1) をリリース。
- デバイスがブートローダーモードでアクティベーション。

デバイスがブートローダーモードになると、ファームウェアは、Microchip (Microchip, SAM-BA Boot Assistant, 2019) からダウンロードできる SAM-BA GUI を使用してプログラミングできます。

## 5 マイクロコントローラーのピンマップ

### 5.1 センサーインターフェース 1

Table 5 センサーインターフェース 1 のマイクロコントローラーピンマップ

信号名	信号グループ	MCU ピン	説明
I2C_SCL_1	I <sup>2</sup> C	PD28	EEPROM の I <sup>2</sup> C インターフェースのクロックピン
I2C_SDA_1	I <sup>2</sup> C	PD27	EEPROM の I <sup>2</sup> C インターフェースのデータピン
S1_SPI_CSN SPI_CSN_1	SPI	PA11	SPI センサーインターフェースのチップセレクトピン
S1_SPI_MISO SPI_MISO_1	SPI	PA12	SPI センサーインターフェースのマスター入力スレーブ出力ピン
S1_SPI_MOSI SPI_MOSI_1	SPI	PA13	SPI センサーインターフェースのマスター出力スレーブ入力ピン
S1_SPI_CLK SPI_CLK_1	SPI	PA14	SPI センサーインターフェースのクロックピン
S1_RST_BGT_RST_1	GPIO	PA0	レベルシフト GPIO、センサーのハードウェアリセットに使用
S1_IRQ_BGT_IRQ_1	GPIO	PC6	レベルシフト GPIO、センサーの割込み要求に使用
S1_GPIO1_GPIO1_1	GPIO	PA9	レベルシフト GPIO、特別な目的はなし
S1_GPIO2_GPIO2_1	GPIO	PA10	レベルシフト GPIO、特別な目的はなし
ADC1_1	アナログ	PE5	ADC チャネル 1、MCU のアナログフロントエンド 0
ADC2_1	アナログ	PC15	ADC チャネル 2、MCU のアナログフロントエンド 1
DAC_1	アナログ	PB13	DAC チャネル、MCU の DAC0
OpenDrain1_1	オープン ドレイン	PD16	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
OpenDrain2_1	オープン ドレイン	PC9	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
OpenDrain3_1	オープン ドレイン	PC28	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
OpenDrain4_1	オープン ドレイン	PA27	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
OpenDrain5_1	オープン ドレイン	PC8	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
OpenDrain_LED_1	オープン ドレイン	PC1	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
S1_SPI_OE	レベル シフター	PD24	SPI 通信用のレベルシフターの Disable ピン - 外部プルダウン抵抗を使用 (高: 無効、低: 有効)
S1_SPI_DIR	レベル シフター	PD11	SPI 通信用レベルシフターの方向ピン - 外部プルアップ抵抗使用 (高: MCU -> センサー、低: センサー -> MCU)
S1_SPI_READ_DIR	レベル シフター	PD17	SPI 通信用レベルシフター (MISO) の方向ピン - 外部プルダウン抵抗使用 (高: MCU -> センサー、低: センサー -> MCU)

信号名	信号グループ	MCU ピン	説明
S1_GPIO_OE	レベルシフター	PD14	GPIO1 および GPIO2 のレベルシフターの Enable ピン - 外部プルアップ抵抗使用 (高: 無効、低: 有効)
S1_GPIO_DIR	レベルシフター	PD18	GPIO1 および GPIO2 のレベルシフターの方向ピン - 外部プルアップ抵抗使用 (高: MCU -> センサー、低: センサー -> MCU)

## 5.2 センサーインターフェース 2

Table 6 センサーインターフェース 2 のマイクロコントローラーピンマップ

信号名	信号グループ	MCU ピン	説明
I2C_SCL_2	I <sup>2</sup> C	PA4	EEPROM の I <sup>2</sup> C インターフェースのクロックピン - Arduino MKR インターフェースと共有
I2C_SDA_2	I <sup>2</sup> C	PA3	EEPROM 用の I <sup>2</sup> C インターフェースのデータピン - Arduino MKR インターフェースと共有
S2_SPI_CLK SPI_CSN_2	SPI	PC25	SPI センサーインターフェースのチップセレクトピン
S2_SPI_MISO SPI_MISO_2	SPI	PC26	SPI センサーインターフェースのマスター入力スレーブ出力ピン
S2_SPI_MOSI SPI_MOSI_2	SPI	PC27	SPI センサーインターフェースのマスター出力スレーブ入力ピン
S2_SPI_CLK SPI_CLK_2	SPI	PC24	SPI センサーインターフェースのクロックピン
S2_RST BGT_RST_2	GPIO	PA17	レベルシフト GPIO-センサーのハードウェアリセットに使用
S2_IRQ BGT_IRQ_2	GPIO	PC11	レベルシフト GPIO-センサーの割込み要求に使用
S2_GPIO1 GPIO1_2	GPIO	PA7	レベルシフト GPIO、特別な目的はなし
S2_GPIO2 GPIO2_2	GPIO	PA2	レベルシフト GPIO、特別な目的はなし
ADC1_2	アナログ	PD30	ADC チャネル 1、MCU のアナログフロントエンド 0
ADC2_2	アナログ	PC13	ADC チャネル 2、MCU のアナログフロントエンド 1
DAC_2	アナログ	PD0	DAC チャネル、MCU の DAC1
OpenDrain1_2	オープンドレイン	PA22	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
OpenDrain2_2	オープンドレイン	PD10	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
OpenDrain3_2	オープンドレイン	PC29	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
OpenDrain4_2	オープンドレイン	PD15	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
OpenDrain5_2	オープンドレイン	PC17	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン

信号名	信号グループ	MCU ピン	説明
OpenDrain_LED_2	オープン ドレイン	PC2	レベルシフトされていない GPIO、3.3 V プッシュプルまたはオープンドレイン
S2_SPI_OE	レベル シフター	PC3	SPI 通信用のレベルシフターの Disable ピン - 外部プルダウン抵抗を使用 (高: 無効、低: 有効)
S2_SPI_DIR	レベル シフター	PA8	SPI 通信用レベルシフターの方向ピン - 外部プルアップ抵抗使用 (高: MCU -> センサー、低: センサー -> MCU)
S2_SPI_READ_DIR	レベル シフター	PD29	SPI 通信用レベルシフター (MISO) の方向ピン - 外部プルダウン抵抗使用 (高: MCU -> センサー、低: センサー -> MCU)
S2_GPIO_OE	レベル シフター	PA1	GPIO1 および GPIO2 のレベルシフターの Enable ピン - 外部プルアップ抵抗使用 (高: 無効、低: 有効)
S2_GPIO_DIR	レベル シフター	PC10	GPIO1 および GPIO2 のレベルシフターの方向ピン - 外部プルアップ抵抗使用 (高: MCU -> センサー、低: センサー -> MCU)

### 5.3 Arduino MKR インターフェース

Table 7 Arduino MKR インターフェースのマイクロコントローラーピンマップ

コネクタ	信号名	MCU ピン	説明
P2.1	5Vusb	-	外部 5 V 電源
P2.2		-	
P2.3	3V3digital	VDDIO	外部 3.3 V 電源
P2.4	グランド	-	グランドライン
P2.5	N_RST	NRST	MCU の反転ハードウェアリセットピン - 外部プルアップ抵抗使用
P2.6	UART_TX	PD26	GPIO、MCU の UART Tx ピン
P2.7	UART_RX	PD25	GPIO、MCU の UART Rx ピン
P2.8	I2C_SCL_2	PA4	GPIO、I <sup>2</sup> C インターフェースのクロックピン - EEPROM のセンサーインターフェース 2 と共有
P2.9	I2C_SDA_2	PA3	GPIO、I <sup>2</sup> C インターフェースのデータピン - EEPROM のセンサーインターフェース 2 と共有
P2.10	SPI0_MISO	PD20	GPIO、SPI センサーインターフェースのマスター入力スレーブ出力ピン
P2.11	SPI0_CLK	PD22	GPIO、SPI センサーインターフェースのクロックピン
P2.12	SPI0_MOSI	PD21	GPIO、SPI センサーインターフェースのマスター出力スレーブ入力ピン
P2.13	SPI0_CSN	PD12	GPIO、SPI センサーインターフェースのチップセレクトピン
P2.14	PWM6	PA5	GPIO、PWM チャネル 1 - ローサイド 3
P3.1	Vref	VREF	MCU のアナログフロントエンドの基準電圧
P3.2	A0	PC0	GPIO、ADC チャネル 0、MCU のアナログフロントエンド 1
P3.3	A1	PC12	GPIO、ADC チャネル 1、MCU のアナログフロントエンド 1
P3.4	A2	PE0	GPIO、ADC チャネル 2、MCU のアナログフロントエンド 1
P3.5	A3	PA19	GPIO、ADC チャネル 3、MCU のアナログフロントエンド 0

コネクタ	信号名	MCU ピン	説明
P3.6	A4	PB3	GPIO、ADC チャネル 4、MCU のアナログフロントエンド 0
P3.7	A5	PE4	GPIO、ADC チャネル 5、MCU のアナログフロントエンド 0
P3.8	A6	PA21	GPIO、ADC チャネル 6、MCU のアナログフロントエンド 0
P3.9	PWM0	PC22	GPIO、PWM チャネル 0 - ローサイド 3
P3.10	PWM1	PC21	GPIO、PWM チャネル 0 - ハイサイド 3
P3.11	PWM2	PC20	GPIO、PWM チャネル 0 - ローサイド 2
P3.12	PWM3	PC19	GPIO、PWM チャネル 0 - ハイサイド 2
P3.13	PWM4	PC18	GPIO、PWM チャネル 0 - ローサイド 1
P3.14	PWM5	PA23	GPIO、PWM チャネル 0 - ハイサイド 0

## 5.4 他のピン

Table 8 他のピンのマイクロコントローラーピンマップ

信号グループ	信号名	MCU ピン	説明
RGB LED	PWMC1_L0_red	PD3	RGB LED 赤色ダイオード、PWM チャネル 1 - ローサイド 0
RGB LED	PWMC1_L1_green	PD5	RGB LED 緑色ダイオード、PWM チャネル 1 - ローサイド 0
RGB LED	PWMC1_L2_blue	PD7	RGB LED 青色ダイオード、PWM チャネル 1 - ローサイド 0
LDO	en_LDO_1	PC30	センサーインターフェース 1 の LDO の有効化
LDO	en_LDO_2	PC31	センサーインターフェース 2 の LDO の有効化

## 6 参考資料

Microchip. (2018, 05). *Atmel Studio 7 User Guide*. Microchip から取得:

<https://www.microchip.com/content/dam/mchp/documents/atmel-start/Atmel-Studio-7-User-Guide.pdf>

Microchip. (2020, 12 20). *SAM-BA boot Assistant*. Microchip Web から取得:

<https://www.microchip.com/en-us/development-tool/SAM-BA-In-system-Programmer>

## 改訂履歴

版数	発行日	変更内容
2.50	2022-02-16	本版は英語版 UG091519 Rader baseboard MCU7 V2.5 について、CYPRESS DEVELOPER COMMUNITY の参画者によって日本語に翻訳されたドキュメントです。
2.70	2023-02-16	これは英語版 UG091519 Rader baseboard MCU7 V2.7 を翻訳した日本語版 V2.7 です。

## Trademarks

All referenced product or service names and trademarks are the property of their respective owners.

**Edition 2023-02-16**

**Published by**

**Infineon Technologies AG**

**81726 Munich, Germany**

**© 2023 Infineon Technologies AG.  
All Rights Reserved.**

**Do you have a question about this  
document?**

**Email:** [erratum@infineon.com](mailto:erratum@infineon.com)

**Document reference**

**UG091519**

## 警告事項

技術的要件に伴い、製品には危険物質が含まれる可能性があります。当該種別の詳細については、インフィニオンの最寄りの営業所までお問い合わせください。

インフィニオンの正式代表者が署名した書面を通じ、インフィニオンによる明示の承認が存在する場合を除き、インフィニオンの製品は、当該製品の障害またはその使用に関する一切の結果が、合理的に人的傷害を招く恐れのある一切の用途に使用することはできないことをご了承ください。