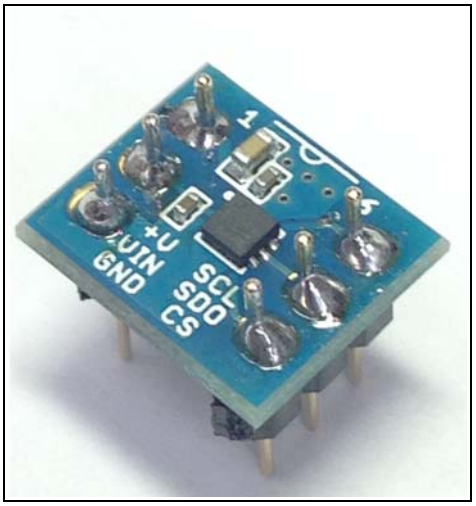
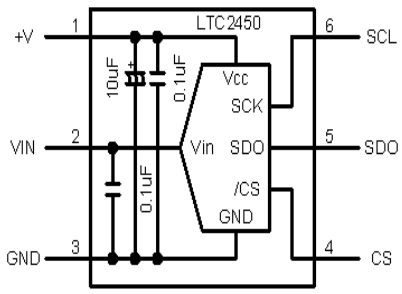
１６ビットΔΣ（デルタシグマ）型Ａ／Ｄコンバータ
リニアテクノロジ社　ＬＴＣ２４５０
２．５４mmピッチ　ＤＩＰ化基板

■特長■

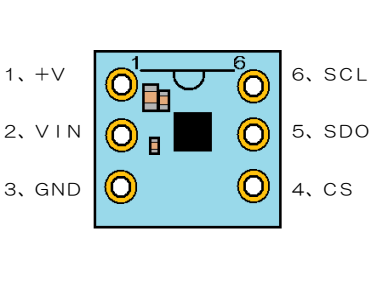
・超小型ＤＦＮパッケージを、使いやすい２．５４mmピッチ６ピンＤＩＰ基板に実装しました。
・入力範囲：ＧＮＤ～Ｖｃｃ（Ｖｃｃ＝２．７Ｖ～５．５Ｖ）
・サンプリング回数：３０回／秒
・入力インピーダンス：１５ｋΩ（ｔｙｐ）
・ＲＭＳ（実効値）ノイズ：０．０２ＬＳＢ
・分解能：１６ビット、ミッシングコード無し＊
　＊ミッシングコード：アナログ入力に対応したデジタルコードの一部が出力されない現象
・オフセット誤差：２ＬＳＢ
・フルスケール誤差：４ＬＳＢ
・多重化アプリケーションに対応する単一変換セトリング時間
・自動シャットダウン付き１サイクル動作
（Ａ／Ｄ変換後、自動的にスリープモードになります）
・消費電流：３００μＡ
・スリープ電流：５０ｎＡ
・内部発振器・外付け部品不要
・２．７Ｖ～５．５Ｖ単一電源動作
・ＳＰＩインタフェース：ＳＣＬ、ＳＤＯ、ＣＳ
（スリープ機能を使用しない場合はＳＣＬ、ＳＤＯの２線式動作が可能です）



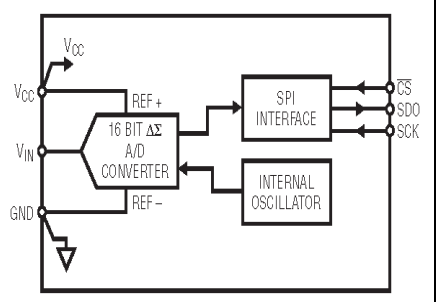
■回路図■



■ピン配置■



■機能ブロック図■



■ピンの名称■

ピン番号	略号	名称	ピン番号	略号	名称
１	＋Ｖ	電源（リファレンスを兼ねます）	６	ＳＣＬ	ＳＰＩシリアルクロック入力
２	ＶＩＮ	アナログ入力電圧	５	ＳＤＯ	ＳＰＩシリアルデータ出力
３	ＧＮＤ	グランド	４	ＣＳ	チップセレクト（アクティブＬ）

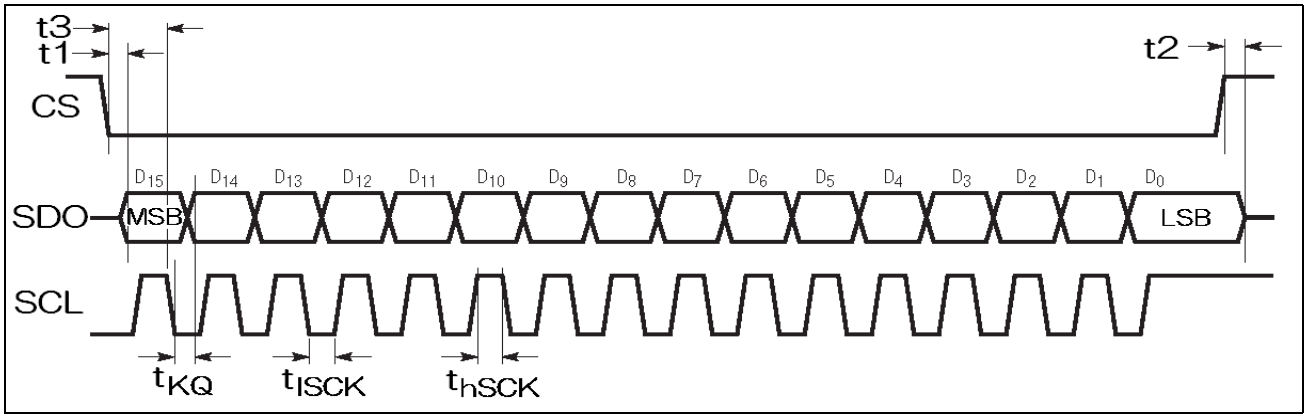
■ピンの説明■

１：＋Ｖ：電源
　正電源電圧およびコンパレータのリファレンス電圧です。
２：ＶＩＮ：アナログ入力電圧
　入力インピーダンスは約１５ｋΩです。サンプリング回数は最大３０回／秒ですので、時間的に変化の激しい入力には適しません。
３：ＧＮＤ：グランド
　アナロググランドとデジタルグランドを兼ねます。なるべく太いラインで配線してください。

４：ＣＳ：チップセレクト
チップ選択デジタル入力です。アクティブロー（Ｌ）。このピンをＬにすると、ＳＤＯデジタル出力がイネーブルされます。このピンをＨにすると、ＳＤＯ出力ピンが高インピーダンス状態になります。
５：ＳＤＯ：ＳＰＩシリアルデータ出力
スリーステートのシリアルデータ出力です。（ＳＤＯはデータ出力ステートの間、シリアルデータ出力に使われ、変換状態のモニタに使う事ができます。モニタ動作の詳細はデバイスのマニュアルを御参照ください）
６：ＳＣＬ：ＳＰＩシリアルクロック入力
ＳＣＬはシリアルデータ出力の同期をとります。デジタルデータが利用可能で（ＡＤＣが変換ステートではない）、ＳＣＫピンに与えられる各立下りエッジに続いて、新しいデータビットがＳＤＯ出力ピンに発生します。

■使い方■

・ＳＰＩ信号を下図のタイミングで制御して、Ａ／Ｄコンバータからデータを取り込みます。
・ＳＤＯはマイコンのＳＤＩと接続してください。
・詳しい使い方はメーカーのデバイスマニュアルをご覧ください。



ｔ１	：ＣＳの立ち下がりエッジからＳＤＯがアクティブになるまでの時間	＝　０秒～最大１００ｎ秒
ｔ２	：ＣＳの立ち上がりエッジからＳＤＯが高インピーダンスになるまでの時間	＝　０秒～最大１００秒
ｔ３	：ＣＳの立ち下がりエッジからＳＣＬの下降エッジまでの時間	＝　最小１００ｎ秒
ｔＫＱ	：ＳＣＬの立ち下がりエッジから次のＳＤＯが有効となるまでの時間	＝　０秒～最大１００ｎ秒
ｔｌＳＣＫ	：ＳＣＬのＬとなっているべき時間	＝　最小２５０ｎ秒
ｔｈＳＣＫ	：ＳＣＬのＨとなっているべき時間	＝　最小２５０ｎ秒

◎ＳＰＩの設定

・アイドル時のクロックがH、クロックの立ち上がりエッジでデータはサンプリングされます。
・データはＭＳＢから送信されます。

★Ａｒｄｕｉｎｏ（UNO）の例

```
unsigned char m_data; //MSB byte
unsigned char l_data;  //LSB byte
const int CS = 10;
void setup() {
    pinMode(CS, OUTPUT);
    SPI.begin();
    SPI.setBitOrder(MSBFIRST);
    SPI.setDataMode(SPI_MODE3);
    SPI.setClockDivider(SPI_CLOCK_DIV16);
}
void loop(){
    digitalWrite(CS.LOW);
    m_data = SPI.transfer(0x00);
    l_data = SPI_transfer(0x00);
    digitalWrite(CS.HIGH);
    //処理コードを書く
    delay(500); //必要に応じてwaitを置く
}
```

★ＰＩＣマイコンの例（ＸＣ８コンパイラ）

```
#include <plib/spi.h>
#include <plib/delays.h>
void main(void){
    char m_data; //MSB byte
    char l_data;  //LSB byte
    OpenSPI( SPI_FOSC_16, MODE_11, SMPMID );
    //SPI_FOSC_16 ←CPUのクロックにより設定
    while(1){
        CS =  0 ;
        m_data = getcSPI();
        l_data = getcSPI();
        CS = 1;
        //処理コードを書く
        _delay_ms(34) //必要に応じてwaitを置く
    }
}
```