

API を使った Windows アプリ開発方法

SDK 内にある API(dll)を使って下の条件の Windows アプリを開発する方法を記します。

◆条件

- ◇開発ツール:Visual Studio 2015
- ◇開発言語:C#(.NET)
- ◇アプリケーション:Windows フォームアプリケーション
- ◇対象 RFID リーダ・ライタ:UC-200f-J2(f ファミリ)、PC とは USB ケーブルで接続

このドキュメントでは、実際にタグ ID を取得するアプリを作成しながら開発手順を記しています。

■注意

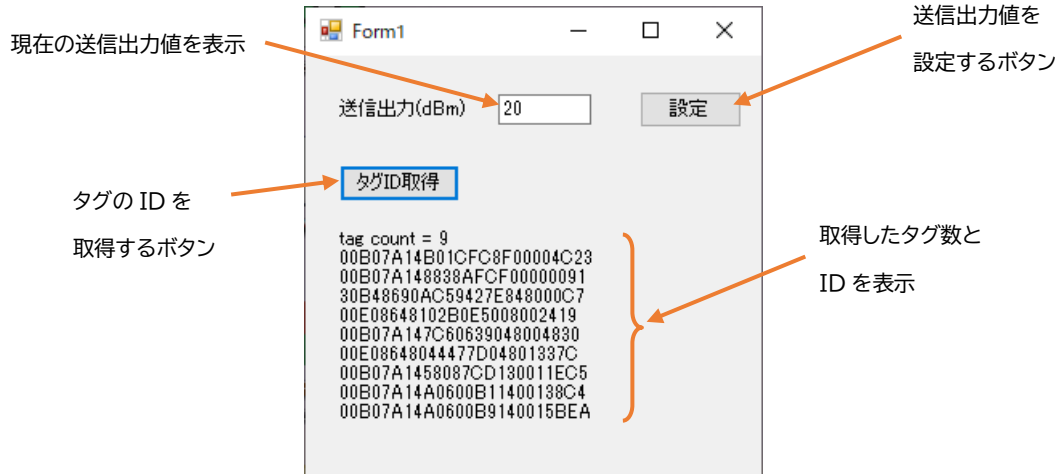
このドキュメント内のソースコードでは、分かりやすさを優先しているためエラー処理を行っていません。

API 関数からの返り値や例外処理など、必要に応じて対処してください。

1. 作成するアプリ

1.1. 概要

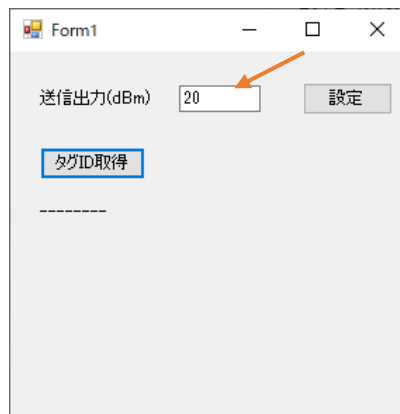
下図はこれから作成するアプリでタグ ID を取得している例です。



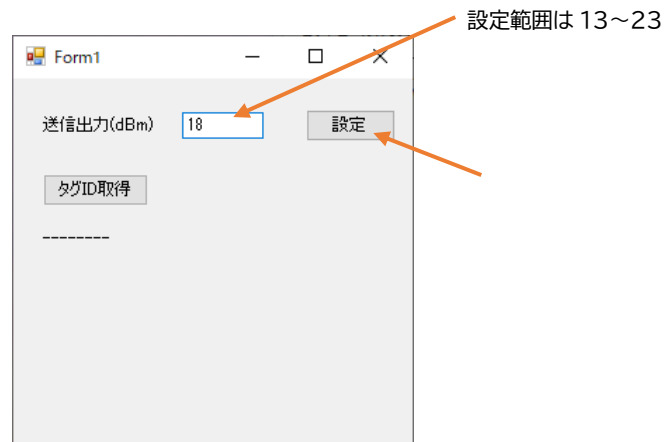
1.2. 操作方法

このアプリの操作方法是下のとおりです。

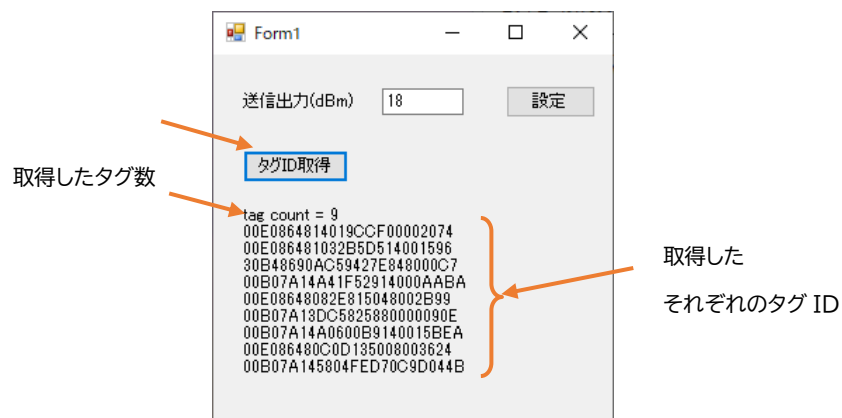
- (1) PC と RFID リーダ・ライタを USB ケーブルで接続します。
- (2) アプリを起動すると、下の2つを実行した後に下図のダイアログが表示されます。
 - ・RFID リーダ・ライタと接続
 - ・現在の送信出力値を取得して表示



- (3) 送信出力値を変更する場合は、テキストボックスに 13~23 の値を入力してから [設定] ボタンをクリックします。



- (4) タグをかざして [タグ ID 取得] ボタンをクリックすると、タグを取得して表示します。



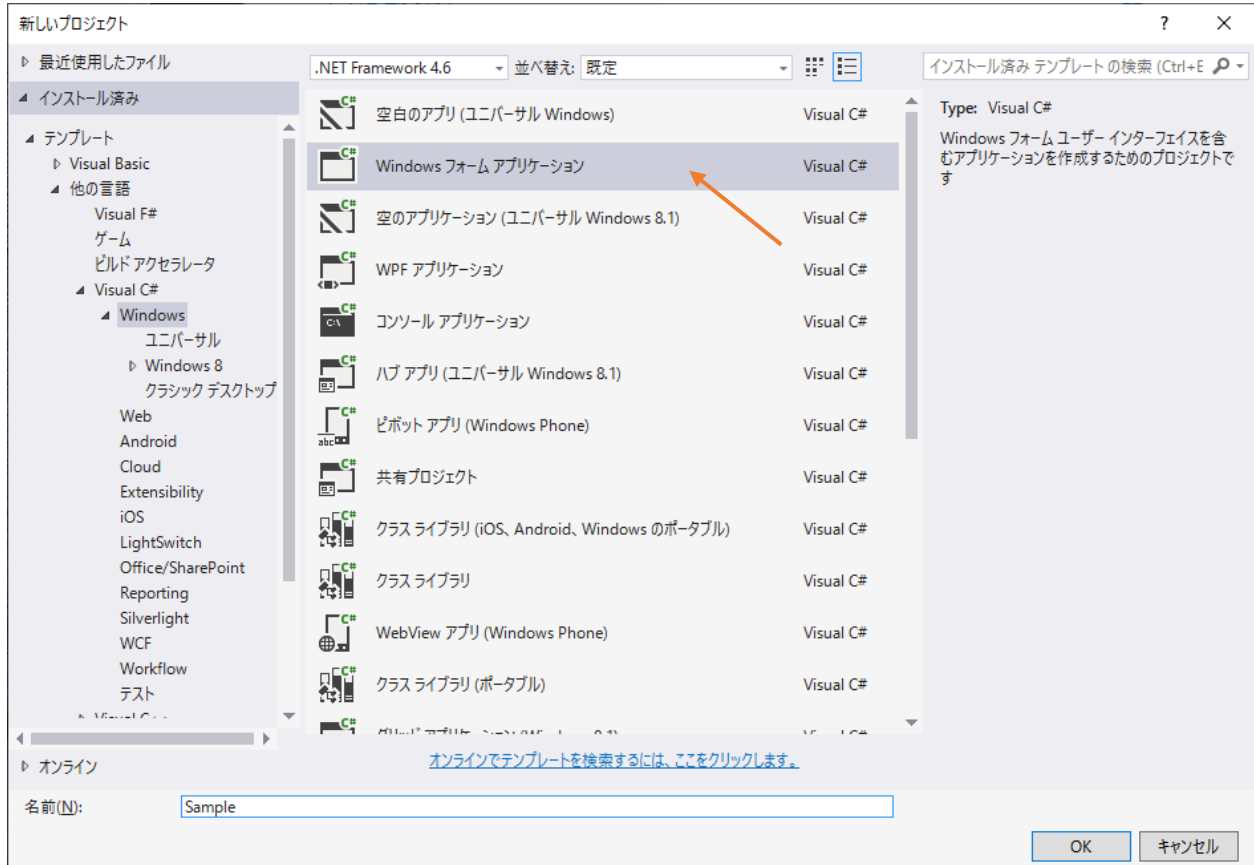
- (5) 画面右上の [×] ボタンでアプリを終了します。
このとき RFID リーダ・ライタと切断します。

それでは、この後実際にこのアプリを作成していきます。

2. アプリの開発

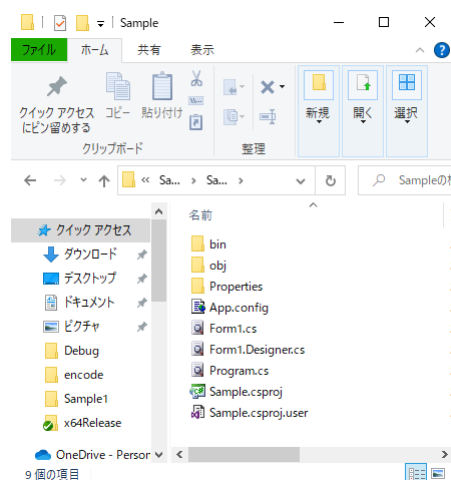
2.1. 新規プロジェクトを作成

Visual Studio を起動して「新しいプロジェクト」を作成します。



[プロジェクト] - [プロパティ] での設定を適宜行います。

一旦[すべて保存]します。保存先をエクスプローラで見ると下のようになっています。



2.2. API をプロジェクトに追加

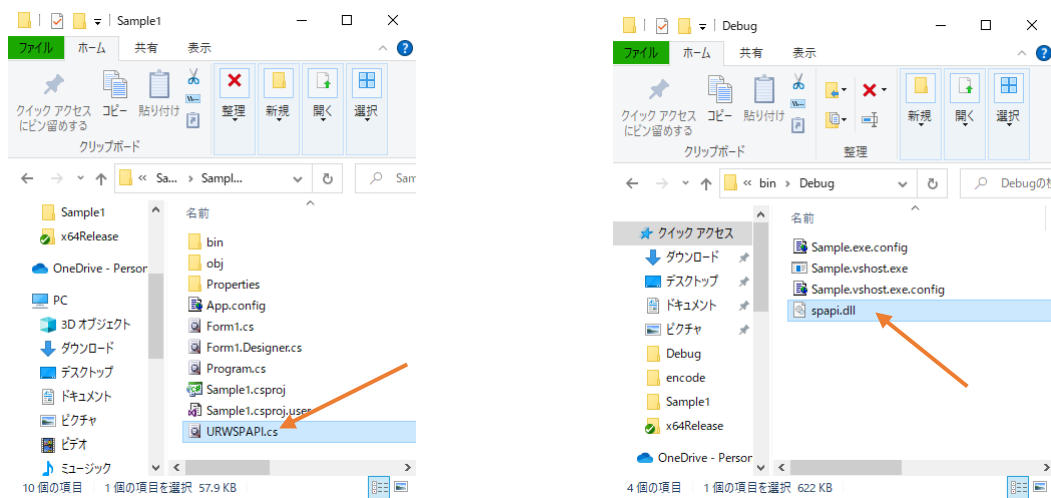
C#での開発で使用する API 関連のファイルは下の2つです。(SDK の中にあります)

- ・URWSPAPI.cs:関数の定義などが記述されているファイルです。
- ・spapi.dll:dll です。

URWSPAPI.cs を左下図のようにアプリケーションのフォルダ内にコピーします。

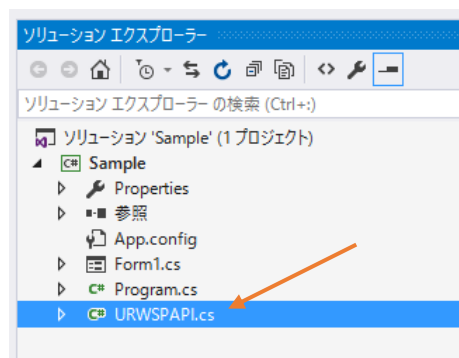
また spapi.dll を右下図のようにアプリケーションの実行ファイル(***.exe)が生成されるフォルダにコピーします。

※spapi.dll は 32bit 用と 64bit 用の2種類あります。生成されるアプリケーションの bit 数により使い分けてください。



URWSPAPI.cs をプロジェクトに追加するために、[プロジェクト]－[既存の項目の追加]でコピーした URWSPAPI.cs を選択します。

下図のように追加されていることを確認してください。



2.3. API を使う準備

Form1.cs を開き API 関数を使うために下図のように

`using spapi;`

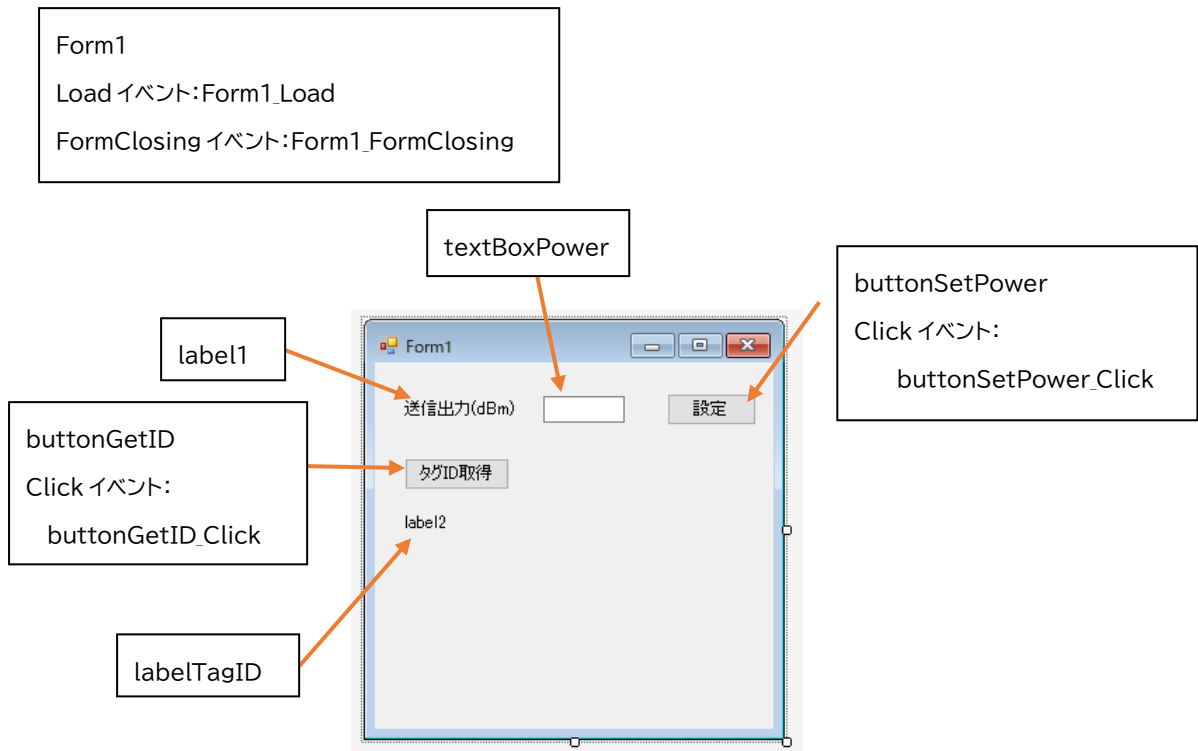
を追記します。

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using spapi; ←

namespace Sample
{
    3 個の参照
    public partial class Form1 : Form
    {
        1 個の参照
        public Form1()
        {
            InitializeComponent();
        }
    }
}
```

2.4. ボタンなどを配置

下図のようにボタンなどを配置します。



2.5. 「接続」と「送信出力値」取得

RFID リーダ・ライタと接続する API 関数は `so_CommSetup()` です。(詳細は API 仕様書を参照)
 このアプリでは Form1 の Load イベント内で接続しています。

下の関数では、f ファミリの RFID リーダ・ライタと USB で接続しています。

```
this.m_iCommNo = URWSPAPI.so_CommSetup(254, 0, 0);
```

これ以降、他の API 関数を呼ぶときに、この関数の戻り値 `m_iCommNo` を第一引数として渡します。

```
namespace Sample
{
    3 個の参照
    public partial class Form1 : Form
    {
        int m_iCommNo = -1;
        1 個の参照
        public Form1()
        {
            InitializeComponent();
        }

        // Form Load イベント
        1 個の参照
        private void Form1_Load(object sender, EventArgs e)
        {
            short PA = 0, DA = 0, Offset = 0;
            this.m_iCommNo = URWSPAPI.so_CommSetup(254, 0, 0);
            URWSPAPI.so_GetTxPower(this.m_iCommNo, ref PA, ref DA, ref Offset);
            textBoxPower.Text = (PA / 100).ToString();
            labelTagID.Text = "-----";
        }
    }
}
```

ここで定義

接続

送信出力値取得

`so_CommSetup()` の引数は、RFID リーダ・ライタのファミリ (f/CB) や接続方法 (USB/RS-232C) により異なります。

また、LAN 接続するときは別関数の `so_CommSetupIP()` を使います。

送信出力値を取得する API 関数は `so_GetTxPower()` です。(詳細は API 仕様書を参照)

```
URWSPAPI.so_GetTxPower(this.m_iCommNo, ref PA, ref DA, ref Offset);
```

第一引数は `so_CommSetup()` からの戻り値です。

第二引数に送信出力値が返ります。100 倍された値を取得するため 100 で割ってから表示します。

2.6. 「切断」

RFID リーダ・ライタと切断する API 関数は `so_CommDelete()` です。

このアプリでは Form1 の FormClosing イベント内で切断しています。

`URWSPAPI.so_CommDelete(this.m_iCommNo);`

引数は `so_CommSetup()` からの戻り値です。

```
// Form Closing イベント
1 個の参照
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    URWSPAPI.so_CommDelete(this.m_iCommNo);
}
```

切断

2.7. 送信出力値「設定」

送信出力値を設定する API 関数は `so_SetTxPAPower()` です。

`URWSPAPI.so_SetTxPAPower(this.m_iCommNo, PA);`

第一引数は `so_CommSetup()` からの戻り値です。

第二引数に送信出力値を指定します。100 倍した値を設定します。

```
// 送信出力「設定」ボタン
1 個の参照
private void buttonSetPower_Click(object sender, EventArgs e)
{
    short PA = (short)(Int16.Parse(textBoxPower.Text) * 100);
    URWSPAPI.so_SetTxPAPower(this.m_iCommNo, PA);
}
```

送信出力値設定

2.8. 「タグ ID 取得」

タグ ID を取得する API 関数として `su_ReadTagID_Single()` を使います。

`URWSPAPI.su_ReadTagID_Single(this.m_iCommNo, tagID, ref count);`

第一引数は `so_CommSetup()` からの返り値です。

第二引数は取得するタグ ID を返す構造体です。

第三引数は取得する最大タグ数を渡し、取得後は実際に取得したタグ数が返ります。

```

// 「タグID取得」ボタン
1 個の参照
private void buttonGetID_Click(object sender, EventArgs e)
{
    URWSPAPI.TypeCTagID_T[] tagID = new URWSPAPI.TypeCTagID_T[10];
    byte count = 10;
    label2.Text = "";

    URWSPAPI.su_ReadTagID_Single(this.m_iCommNo, tagID, ref count);
    string str = "";
    str += string.Format("tag count = {0:d}¥r¥n", count);

    for (int cnt = 0; cnt < count; cnt++)
    {
        for (int i = 0; i < tagID[cnt].EPC_Length; i++)
        {
            str += string.Format("{0:X2}", tagID[cnt].EPC[i]);
        }
        str += "¥r¥n";
    }
    label2.Text = str;
}
    
```

10 個分を用意

タグ ID 取得

取得したタグ数を表示

取得したタグ ID を表示

以上