

SavoySecsI ActiveX コントロール  
ユーザーガイド

## 1 改訂履歴

バージョン	日付	氏名	説明
1.00	2009年7月31日	Hikaru Okada	新規に作成。
1.00a	2009年8月22日	Hikaru Okada	マニュアルのページ数が大きくなったため分割。
1.00b	2009年8月30日	Hikaru Okada	一部の誤記を訂正。
1.00c	2010年7月22日	Hikaru Okada	一部の誤記を訂正。

---

---

## 2 目次

1	改訂履歴 .....	2
2	目次 .....	3
3	SavoySecsl .....	4
3.1	プロパティ .....	5
3.1.1	Appearance .....	5
3.1.2	BaudRate .....	6
3.1.3	BorderStyle .....	7
3.1.4	Connect .....	8
3.1.5	DeviceID .....	9
3.1.6	IniFileName .....	10
3.1.7	IniSection .....	11
3.1.8	Log .....	12
3.1.9	LogBakCount .....	13
3.1.10	LogFileName .....	14
3.1.11	LogSize .....	15
3.1.12	LogVerbose .....	16
3.1.13	Master .....	17
3.1.14	MSEC .....	18
3.1.15	PortNumber .....	19
3.1.16	Retry .....	20
3.1.17	T1 .....	21
3.1.18	T2 .....	22
3.1.19	T3 .....	23
3.1.20	T4 .....	24
3.2	メソッド .....	25
3.2.1	AboutBox .....	25
3.2.2	LoadIniFile .....	26
3.2.3	Send .....	27
3.2.4	Setup .....	28
3.3	イベント .....	30
3.3.1	Problem .....	30
3.3.2	Received .....	31
3.3.3	Sent .....	32

### 3 SavoySecsI

SavoySecsI コントロールは SEMI E4 (SECS-I)の機能を作成するための開発支援製品です。装置側ソフト、ホスト側ソフトのいずれの開発にも使えます。SavoySecsII コントロールと組み合わせて使います。

#### プロパティ

名前	説明
Appearance	外観を決定する値を取得または設定します。
BaudRate	シリアルケーブル上での通信速度を取得または設定します。
BorderStyle	境界線スタイルを取得または設定します。
Connect	SECS-I 接続状態を取得または設定します。
DeviceID	デバイス ID を取得または設定します。
IniFileName	設定を読み書きするための INI ファイル名を取得または設定します。
IniSection	設定を読み書きするための INI ファイル中のセクション名を取得または設定します。
Log	ログの有効・無効を取得または設定します。
LogBakCount	ログのバックアップファイルの数を取得または設定します。
LogFileName	ログファイル名を取得または設定します。
LogSize	ログファイルのサイズを取得または設定します。
LogVerbose	詳細部分のログが有効かどうかを取得または設定します。
Master	マスターかスレーブのどちらの役割で動作するかを取得または設定します。
MSEC	MSEC(三菱 SECS プロトコル)が有効かどうかを取得または設定します。
PortNumber	シリアルポート番号を取得または設定します。
Retry	送信エラーが発生した場合に何回リトライするかを取得または設定します。
T1	T1 タイムアウトを取得または設定します。
T2	T2 タイムアウトを取得または設定します。
T3	T3 タイムアウトを取得または設定します。
T4	T4 タイムアウトを取得または設定します。

#### メソッド

名前	説明
AboutBox	バージョン情報を表示します。
LoadIniFile	設定内容を INI ファイルから読み出します。
Send	指定された SECS-II メッセージを送信します。
Setup	セットアップ画面を表示します。

#### イベント

名前	説明
Problem	エラーが発生したときに通知されます。
Received	SECS-II メッセージを受信したときに通知されます。
Sent	SECS-II メッセージが送信されたときに通知されます。

## 3.1 プロパティ

### 3.1.1 Appearance

SavoySecsI コントロールの外観を決定する値を取得または設定します。

値	説明
0	フラット
1	凹んだ枠線

#### 構文

Visual Basic 6.0

```
Appearance As Integer
```

Visual C++ 6.0

```
short GetAppearance()  
void SetAppearance(short)
```

#### 使用例

Visual Basic 6.0

```
.Appearance = 0 ' flat  
.Appearance = 1 ' sunken
```

Visual C++ 6.0

```
m_ctrl.SetAppearance(0); // flat  
m_ctrl.SetAppearance(1); // sunken
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.2 BaudRate

シリアルケーブル上での通信速度を取得または設定します。

値	説明
300	300 ビット/秒
600	600 ビット/秒
1200	1200 ビット/秒
2400	2400 ビット/秒
4800	4800 ビット/秒
9600	9600 ビット/秒
19200	19200 ビット/秒
38400	38400 ビット/秒
76800	76800 ビット/秒

デフォルトは 9600bps です。

#### 構文

Visual Basic 6.0

```
BaudRate As Long
```

Visual C++ 6.0

```
long GetBaudRate()
void SetBaudRate(long)
```

#### 使用例

Visual Basic 6.0

```
.PortNumber = 0
.BaudRate = 9600
.Retry = 3
.Connect = True
```

Visual C++ 6.0

```
m_ctrl.SetPortNumber(0);
m_ctrl.SetBaudRate(9600);
m_ctrl.SetRetry(3);
m_ctrl.SetConnect(true);
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.3 BorderStyle

SavoySecsI コントロールの境界線スタイルを取得または設定します。

値	説明
0	境界線なし
1	境界線あり

#### 構文

Visual Basic 6.0

```
BorderStyle As Integer
```

Visual C++ 6.0

```
short GetBorderStyle()  
void SetBorderStyle(short)
```

#### 使用例

Visual Basic 6.0

```
.BorderStyle = 0 ' no border  
.BorderStyle = 1 ' with border
```

Visual C++ 6.0

```
m_ctrl.SetBorderStyle(0); // no border  
m_ctrl.SetBorderStyle(1); // with border
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.4 Connect

SECS-I 接続状態を取得または設定します。もし Connect プロパティが True の場合、SavoySecsI コントロールは指定されたシリアルポートをオープンしようと試みます。もしシリアルポートがオープンできなかった場合、Connect プロパティは False にセットされます。

#### 構文

Visual Basic 6.0

```
Connect As Boolean
```

Visual C++ 6.0

```
BOOL GetConnect()  
void SetConnect(BOOL)
```

#### 使用例

Visual Basic 6.0

```
.PortNumber = 0  
.BaudRate = 9600  
.Retry = 3  
.Connect = True
```

Visual C++ 6.0

```
m_ctrl.SetPortNumber(0);  
m_ctrl.SetBaudRate(9600);  
m_ctrl.SetRetry(3);  
m_ctrl.SetConnect(true);
```

#### 特記事項

#### 参照



### 3.1.5 DeviceID

デバイス ID を取得または設定します。デバイス ID は SECS-II ヘッダの先頭 2 ビット目から 15 ビットです。

SECS-I では下記のヘッダ構造が使用されます。

Byte	説明	
1	R	デバイス ID
2		
3	W	ストリーム番号
4		ファンクション番号
5	E	ブロック番号
6		
7		ソース ID
8		
9		トランザクション ID
10		

#### 構文

Visual Basic 6.0

```
DeviceID As Long
```

Visual C++ 6.0

```
long GetDeviceID()
void SetDeviceID(long)
```

#### 使用例

Visual Basic 6.0

```
.DeviceID = 1
```

Visual C++ 6.0

```
m_ctrl.SetDeviceID(1);
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.6 IniFileName

設定を読み書きするための INI ファイル名を取得または設定します。もし INI ファイル名がフルパス名かフォルダの相対参照を含む場合は、INI ファイルは指定された場所に作成されます。そうでない場合は、Windows のシステムフォルダに INI ファイルが作成されます。この理由からフォルダ名と共に使用するのが推奨されます。もしカレントディレクトリが INI ファイルの場所であれば、「./」を先頭に付けます。

「/」または「¥」をフォルダ名の区切り文字として使用できます。

#### 構文

Visual Basic 6.0

```
IniFileName As String
```

Visual C++ 6.0

```
CString GetIniFileName()  
void SetIniFileName(LPCTSTR)
```

#### 使用例

Visual Basic 6.0

```
.IniFileName = "./Savoy.ini"
```

Visual C++ 6.0

```
m_ctrl.SetIniFileName("./Savoy.ini");
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.7 IniSection

設定を読み書きするための INI ファイル中のセクション名を取得または設定します。

#### 構文

Visual Basic 6.0

```
IniSection As String
```

Visual C++ 6.0

```
CString GetIniSection()  
void SetIniSection(LPCTSTR)
```

#### 使用例

Visual Basic 6.0

```
.IniSection = "Host" ' [Host] section
```

Visual C++ 6.0

```
m_ctrl.IniSection("Host"); // [Host] section
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.8 Log

ログの有効・無効を取得または設定します。もしこのプロパティが true の場合、処理情報はログファイルに書き込まれます。もしこのプロパティが false の場合はログファイルには記録されません。

値	説明
True	ログファイルに記録する
False	ログファイルに記録しない

#### 構文

Visual Basic 6.0

```
Log As Boolean
```

Visual C++ 6.0

```
BOOL GetLog()  
void SetLog(BOOL)
```

#### 使用例

Visual Basic 6.0

```
.Log = True
```

Visual C++ 6.0

```
m_ctrl.SetLog(true);
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.9 LogBakCount

ログのバックアップファイルの数を取得または設定します。もしログファイルの実際のファイルサイズが LogSize プロパティを超えた場合、SavoySecsI コントロールはファイル名をリネームし、新たに空のログファイルを作成します。もしバックアップファイルの数が LogBakCount プロパティに達した場合、SavoySecsI コントロールは古いバックアップファイルから消去します。

#### 構文

Visual Basic 6.0

```
LogBakCount As Integer
```

Visual C++ 6.0

```
short GetLogBakCount()  
void SetLogBakCount(short)
```

#### 使用例

Visual Basic 6.0

```
.LogBakCount = 10
```

Visual C++ 6.0

```
m_ctrl.SetLogBakCount(10);
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.10 LogFileName

ログファイル名を取得または設定します。

#### 構文

Visual Basic 6.0

```
LogFileName As String
```

Visual C++ 6.0

```
CString GetLogFileName()  
void SetLogFileName(LPCTSTR)
```

#### 使用例

Visual Basic 6.0

```
.LogFileName = "Savoy"
```

Visual C++ 6.0

```
m_ctrl.SetLogFileName("Savoy");
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.11 LogSize

ログファイルのサイズを取得または設定します。もしログファイルの実際のファイルサイズが LogSize プロパティを超えた場合、SavoySecsI コントロールはファイル名をリネームし、新たに空のログファイルを作成します。もしバックアップファイルの数が LogBakCount プロパティに達した場合、SavoySecsI コントロールは古いバックアップファイルから消去します。

#### 構文

Visual Basic 6.0

```
LogSize As Long
```

Visual C++ 6.0

```
long GetLogSize()  
void SetLogSize(long)
```

#### 使用例

Visual Basic 6.0

```
.LogSize = 1024
```

Visual C++ 6.0

```
m_ctrl.SetLogSize(1024);
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.12 LogVerbose

詳細部分のログが有効かどうかを取得または設定します。

値	説明
True	詳細ログが有効
False	詳細ログを記録しない

#### 構文

Visual Basic 6.0

```
LogVerbose As Boolean
```

Visual C++ 6.0

```
BOOL GetLogVerbose()  
void SetLogVerbose(BOOL)
```

#### 使用例

Visual Basic 6.0

```
.LogVerbose = False
```

Visual C++ 6.0

```
m_ctrl.SetLogVerbose(false);
```

#### 特記事項

永続化プロパティ。

#### 参照



### 3.1.13 Master

SavoySecsI コントロールがマスターかスレーブのどちらの役割で動作するかを取得または設定します。SEMI の E4 スタンドは装置側をマスター、ホスト側をスレーブにするよう定義しています。

値	説明
False	スレーブ
True	マスター

#### 構文

```
Visual Basic 6.0
Master As Boolean
```

```
Visual C++ 6.0
BOOL GetMaster()
void SetMaster(BOOL)
```

#### 使用例

```
Visual Basic 6.0
.Master = True ' Equipment
.Master = False ' Host
```

```
Visual C++ 6.0
m_ctrl.SetMaster(true); ' Equipment
m_ctrl.SetMaster(false); ' Host
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.14 MSEC

MSEC(三菱 SECS プロトコル)が有効かどうかを取得または設定します。

値	説明
False	SECS-I
True	MSEC

#### 構文

Visual Basic 6.0

```
MSEC As Boolean
```

Visual C++ 6.0

```
BOOL GetMsec()  
void SetMsec(BOOL)
```

#### 使用例

Visual Basic 6.0

```
.MSEC = True
```

Visual C++ 6.0

```
m_ctrl.SetMsec(true);
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.15 PortNumber

シリアルポート番号を取得または設定します。ポート番号は 0 から始まります。例えば comm #1 は 0、comm #2 は 1 といった具合です。

デフォルトは 0 です。

#### 構文

Visual Basic 6.0

```
PortNumber As Integer
```

Visual C++ 6.0

```
short GetPortNumber()  
void SetPortNumber(short)
```

#### 使用例

Visual Basic 6.0

```
.PortNumber = 0  
.BaudRate = 9600  
.Retry = 3  
.Connect = True
```

Visual C++ 6.0

```
m_ctrl.SetPortNumber(0);  
m_ctrl.SetBaudRate(9600);  
m_ctrl.SetRetry(3);  
m_ctrl.SetConnect(true);
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.16 Retry

送信エラーが発生した場合に何回リトライするかを取得または設定します。

#### 構文

Visual Basic 6.0

```
Retry As Integer
```

Visual C++ 6.0

```
short GetRetry()  
void SetRetry(short)
```

#### 使用例

Visual Basic 6.0

```
.PortNumber = 0  
.BaudRate = 9600  
.Retry = 3  
.Connect = True
```

Visual C++ 6.0

```
m_ctrl.SetPortNumber(0);  
m_ctrl.SetBaudRate(9600);  
m_ctrl.SetRetry(3);  
m_ctrl.SetConnect(true);
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.17 T1

SECS-I の T1 タイムアウトを取得または設定します。単位はミリ秒。

#### 構文

Visual Basic 6.0

```
T1 As Long
```

Visual C++ 6.0

```
long GetT1()  
void SetT1(long)
```

#### 使用例

Visual Basic 6.0

```
Dim IT1 As Long  
IT1 = .T1
```

Visual C++ 6.0

```
long IT1 = m_ctrl.GetT1();
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.18 T2

SECS-I の T2 タイムアウトを取得または設定します。単位はミリ秒。

#### 構文

Visual Basic 6.0

```
T2 As Long
```

Visual C++ 6.0

```
long GetT2()  
void SetT2(long)
```

#### 使用例

Visual Basic 6.0

```
Dim IT2 As Long  
IT2 = .T2
```

Visual C++ 6.0

```
long IT2 = m_ctrl.GetT2();
```

#### 特記事項

永続化プロパティ。

#### 参照

### 3.1.19 T3

SECS-I の T3 タイムアウトを取得または設定します。単位はミリ秒。

#### 構文

Visual Basic 6.0

```
T3 As Long
```

Visual C++ 6.0

```
long GetT3()  
void SetT3(long)
```

#### 使用例

Visual Basic 6.0

```
Dim IT3 As Long  
IT3 = .T3
```

Visual C++ 6.0

```
long IT3 = m_ctrl.GetT3();
```

#### 特記事項

永続化プロパティ。

#### 参照

## 3.1.20 T4

SECS-I の T4 タイムアウトを取得または設定します。単位はミリ秒。

**構文**

Visual Basic 6.0

```
T4 As Long
```

Visual C++ 6.0

```
long GetT4()  
void SetT4(long)
```

**使用例**

Visual Basic 6.0

```
Dim IT4 As Long  
IT4 = .T4
```

Visual C++ 6.0

```
long IT4 = m_ctrl.GetT4();
```

**特記事項**

永続化プロパティ。

**参照**



## 3.2 メソッド

### 3.2.1 AboutBox

バージョン情報を表示します。

#### 構文

```
Visual Basic 6.0
```

```
Sub AboutBox()
```

```
Visual C++ 6.0
```

```
void AboutBox()
```

#### 戻り値

ありません。

#### 使用例

```
Visual Basic 6.0
```

```
.AboutBox
```

```
Visual C++ 6.0
```

```
m_ctrl.AboutBox();
```

#### 特記事項

#### 参照

### 3.2.2 LoadIniFile

設定内容を INI ファイルから読み出し、プロパティを初期化します。読み込みができない場合はリソースに埋め込まれている値がセットされます。

このメソッドは Setup メソッドで保存されたプロパティを取り出すためアプリケーションの起動時に呼び出すといいでしょう。

#### 構文

Visual Basic 6.0

```
Function LoadIniFile() As Boolean
```

Visual C++ 6.0

```
BOOL LoadIniFile()
```

#### 戻り値

正しく初期化できた場合は True が、失敗した場合は False が返ります。失敗するケースとしては、IniFileName プロパティと IniSection プロパティが正しくない場合が考えられます。

#### 使用例

Visual Basic 6.0

```
Private Sub Form_Load()  
    SwingHsms1.LoadIni  
End Sub
```

Visual C++ 6.0

```
void CxxxView::OnInitialUpdate()  
{  
    ...  
    m_ctrl.LoadIni();  
    ...  
}
```

#### 特記事項

#### 参照

### 3.2.3 Send

指定された SECS-II メッセージを送信します。

#### 構文

Visual Basic 6.0
Function Send(IpszMessage As String) As Boolean

Visual C++ 6.0
BOOL Send(LPCTSTR IpszMessage)

引数	説明
IpszMessage	送信するメッセージ。

#### 戻り値

このメソッドは常に True を返します。

#### 使用例

Visual Basic 6.0
SavoyHsms1.Send SavoySecsII1.Msg

Visual C++ 6.0
m_ctrl.Send(m_msg.GetMsg());

#### 特記事項

Send メソッドは送信キューにメッセージをセットします。Send メソッドから戻っても実際の送信が完了する訳ではないことに注意してください。送信の完了は Sent イベントで通知されます。

送信に失敗した場合はリトライを繰り返し、その後 Problem イベントでエラーコード 8(リトライオーバー)で、送信に失敗したことが通知されます。

#### 参照

### 3.2.4 Setup

セットアップ画面を表示します。パラメータを変更して OK ボタンを押すと INI ファイルに書き込まれます。

#### SECS タブ

項目	説明
Comm port #	シリアルポート番号
Baud rate	通信速度
Retry	リトライ回数
Master	マスター(装置)か、スレーブ(ホスト)かの選択
Device ID	デバイス ID
MSEC	三菱 SECS が有効かの選択
T1	T1 タイマー
T2	T2 タイマー
T3	T3 タイマー
T4	T4 タイマー

#### Log タブ

項目	説明
Enable logging	ログに記録する場合はチェックをつけます。

Enable communication log	詳細の通信ログを記録する場合はチェックをつけます。
File name	ログファイル名。
Number of backup files	バックアップファイルの数。
Maximum size of each file	ログファイルのサイズ。

### 構文

Visual Basic 6.0
Function Setup(lpszCaption As String) As Boolean

Visual C++ 6.0
BOOL Setup(LPCTSTR lpszCaption)

引数	説明
lpszCaption	ダイアログボックスのキャプションタイトル。もしこの値が NULL または空文字列の場合、IniSection プロパティがキャプションタイトルとして使われます。

### 戻り値

ユーザが OK ボタンを押し、正しく INI ファイルに保存されると True を返します。ユーザが Cancel ボタンを押したり、INI ファイルの保存に失敗すると False を返します。

### 使用例

Visual Basic 6.0
.Setup "SECS-I Configuration"

Visual C++ 6.0
m_ctrl.Setup("SECS-I Configuration");

### 特記事項

### 参照

### 3.3 イベント

#### 3.3.1 Problem

エラーが発生したときに通知されます。

#### 構文

Visual Basic 6.0

```
Event Problem(sErrorCode As Integer, lpszAdditionalInfo As String)
```

Visual C++ 6.0

```
void OnProblem(short sErrorCode, LPCTSTR lpszAdditionalInfo)
```

引数	説明
sErrorCode	エラーコード。エラーコードは以下のいずれかの値です。
lpszAdditionalInfo	追加情報。 エラーコード 8 (リトライオーバー) の時には、送信に失敗したメッセージとなります。

#### 使用例

Visual Basic 6.0

```
Text1.Text = "Error - Code : " + Format$(sErrorCode)
```

Visual C++ 6.0

```
TRACE("Error - Code : %d",sErrorCode);
```

#### 特記事項

エラーコード	列挙型	説明
1	ErrorSecsUnexpectedChar	想定していない文字を受信
2	ErrorSecsBadLength	長さが異常
3	ErrorSecsBadSum	チェックサムが異常
4	ErrorSecsBadParity	パリティエラー
5	ErrorSecsBadMultiBlockNumber	マルチブロックのブロック番号が異常
6	ErrorSecsBadBlockNumber	ブロック番号が異常
7	ErrorSecsNak	NAK を受信
8	ErrorSecsRetry	リトライオーバー
9	ErrorSecsT4	T4 タイムアウト
10	ErrorSecsT2	T2 タイムアウト
11	ErrorSecsT1	T1 タイムアウト
12	ErrorSecsT3	T3 タイムアウト
13	ErrorSecsBadDeviceID	デバイス ID が不一致

#### 参照

### 3.3.2 Received

SavoySecsI コントロールがシリアルポート経由で SECS-II メッセージを受信したときに通知されます。

#### 構文

Visual Basic 6.0

```
Event Received(lpszMsg As String)
```

Visual C++ 6.0

```
void OnReceived(LPCTSTR lpszMsg)
```

引数	説明
lpszMsg	受信したメッセージ

#### 使用例

Visual Basic 6.0

```
With SavoySecsII1
  .Msg = lpszMsg
  If .Stream = 1 AND .Function = 13 Then
    ' S1F13
    ...
  End If
End With
```

Visual C++ 6.0

```
m_msg.SetMsg(lpszMsg);
if(m_msg.GetStream()==1 && m_msg.GetFunction()==13)
{
  // S1F13
  ...
}
```

#### 特記事項

#### 参照

### 3.3.3 Sent

SECS-II メッセージが送信されたときに通知されます。

#### 構文

Visual Basic 6.0

```
Event Sent(IpszMsg As String)
```

Visual C++ 6.0

```
void OnSent(LPCTSTR IpszMsg)
```

引数	説明
IpszMsg	送信されたメッセージ

#### 使用例

Visual Basic 6.0

```
SavoySecsI1.Msg = IpszMsg
```

Visual C++ 6.0

```
m_msg.SetMsg(IpszMsg);
```

#### 特記事項

#### 参照