LR5-LAN ソケット通信
サンプルプログラム
（Windows VB.NET）

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

内容

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

CONFIDENTIAL

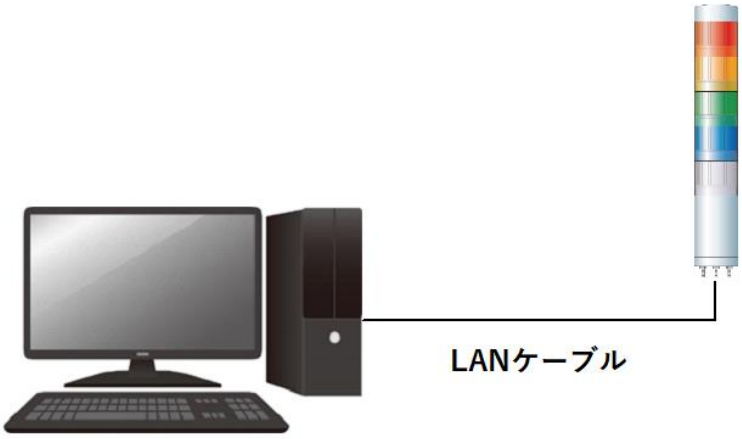
PATLITE Corporation 2025
All rights reserved

1. 概要

LR5-LAN をソケット通信で制御するための、サンプルプログラムの概要を記載する。
本プログラムは、パトライトが提供する DLL を使用せずに Microsoft Visual Basic .NET での制御をおこなうことを目的としている。

1.1. システム概要

本プログラムのシステム構成図は以下の通り。
本プログラムでは、1 台の LR5-LAN の機器をソケット通信で制御を行う。



2. 開発環境

サンプルプログラムの開発環境を以下に示す。

開発環境		備考
開発 OS	Windows11 64bit	
開発言語	VB.NET	.Net Framework 4.5 以降
アプリ種別	CUI アプリケーション	
開発ツール	VisualStudio2022 Professional	

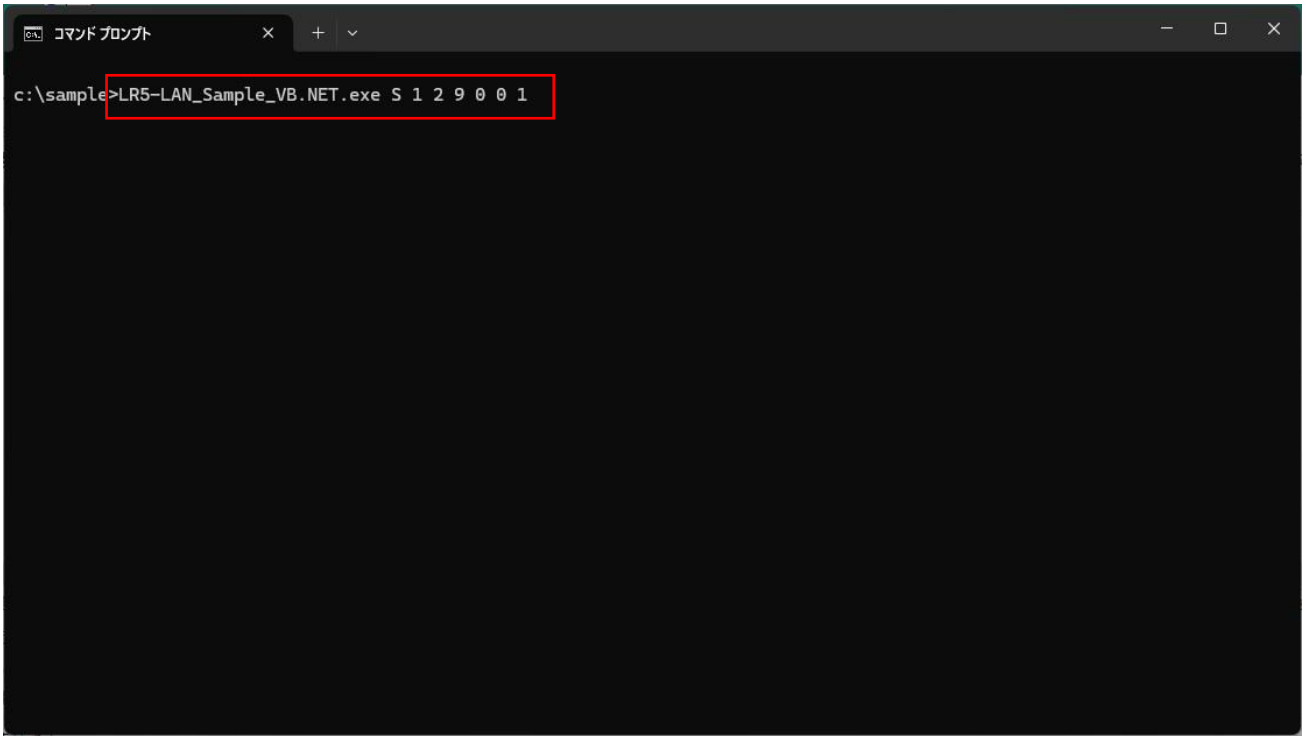
CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3. アプリケーション概要

3.1. コマンド操作説明

コマンドプロンプト上では、ビルド時に作成したLR5-LAN_Sample_VB.NET.exeのファイルの場所に移動して、コマンドライン引数を指定することで各動作のコマンドを実行される。



3.1.1. コマンド一覧

コマンド名	内容
動作制御コマンド	LED ユニットの各色のパターンとブザー(吹鳴・停止)を制御する
クリアコマンド	LED ユニットを消灯し、ブザーを停止する
状態取得コマンド	LED ユニットおよびブザーの状態を取得する

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.1.2. 動作制御コマンド

以下のコマンドライン引数を指定して、コマンドを実行する

No.	コマンドライン引数	値
1	コマンド ID	S
2	LEDユニット赤	消灯:0 点灯:1 点滅（低速):2 点滅（中速):3 点滅（高速):4 シングルフラッシュ:5 ダブルフラッシュ:6 トリプルフラッシュ:7 変化なし:9
3	LEDユニット黄	
4	LEDユニット緑	
5	LEDユニット青	
6	LEDユニット白	
7	ブザーパターン	停止:0 吹鳴:1 変化なし:9

例:LR5-LAN_Sample_VB.NET.exe S 1 2 9 0 0 1

3.1.3. クリアコマンド

以下のコマンドライン引数を指定して、コマンドを実行する

No.	コマンドライン引数	値
1	コマンド ID	C

例:LR5-LAN_Sample_VB.NET.exe C

3.1.4. 状態取得コマンド

以下のコマンドライン引数を指定して、コマンドを実行する

No.	コマンドライン引数	値
1	コマンド ID	G

例:LR5-LAN_Sample_VB.NET.exe G

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.2. 関数説明

3.2.1. 関数一覧

関数名	説明
SocketOpen	LR5-LAN に接続する
SocketClose	ソケットをクローズする
SendCommand	コマンドを送信する
PNS_RunControlCommand	PNS コマンドの動作制御コマンド送信する
PNS_ClearCommand	PNS コマンドのクリアコマンド送信する
PNS_GetDataCommand	PNS コマンドの状態取得コマンド送信する

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.2.2. LR5-LAN に接続

関数名	Public Function SocketOpen(ByVal ip As String, ByVal port As Integereger) As Integereger	
パラメータ	ByVal ip As String	LR5-LAN の IP アドレス
	ByVal port As Integereger	LR5-LAN のポート番号
戻り値	Integer	成功:0、失敗:0 以外
説明	指定した IP アドレスとポート番号の LR5-LAN にソケット通信で接続する	

関数の使用方法	' Socket クラスの変数を定義
	Private sock As Socket = Nothing
	
	' メイン関数
	Sub Main()
	' LR5-LAN に接続
	Dim ret As Integer
	ret = SocketOpen("192.168.10.1", 10000)
	If ret = -1 Then
	Return
	End If
	End Sub	

備考	プログラムの概要は「4.1LR5-LAN に接続」を参照	

3.2.3. ソケットをクローズ

関数名	Public Sub SocketClose()
パラメータ	なし
戻り値	なし
説明	LR5-LAN に接続したソケットをクローズする

関数の使用方法	' メイン関数
	Sub Main()
	' LR5-LAN に接続
	Dim ret As Integer
	ret = SocketOpen("192.168.10.1", 10000)
	If ret = -1 Then
	Return
	End If
	
	' ソケットをクローズ
	SocketClose()
	End Sub

備考	プログラムの概要は「4.2 ソケットをクローズ」を参照

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.2.4. コマンドを送信

関数名	Public Function SendCommand(ByVal sendData As Byte(), ByRef recvData As Byte()) As Integer	
パラメータ	ByVal sendData As Byte()	送信データ
	ByRef recvData As Byte()	受信データ
戻り値	Integer	成功:0、失敗:0 以外
説明	接続した LR5-LAN にデータを送信して、応答データを返す	

関数の使用方法	' メイン関数
	Sub Main()
	　' LR5-LAN に接続
	　Dim ret As Integer
	　ret = SocketOpen("192.168.10.1", 10000)
	　If ret = -1 Then
	　　Return
	　End If
	
	　' 送信データを作成
	　Dim sendData As Byte(7)
	　Dim recvData As Byte()
	　sendData(0) = &H41
	　sendData(1) = &H42
	　sendData(2) = &H53
	　sendData(3) = &H0
	　sendData(4) = &H0
	　sendData(5) = &H0
	　sendData(6) = &H1
	
	　' コマンドを送信
	　ret = SendCommand(sendData, recvData)
	　If ret <> 0 Then
	　　Debug.WriteLine("failed to send data")
	　　Return -1
	　End If
	
	　' ソケットをクローズ
	　SocketClose()
	End Sub	

備考	プログラムの概要は「4.3 コマンドを送信」を参照	

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.2.5. PNS コマンドの動作制御コマンド送信

関数名	Public Function PNS_RunControlCommand(ByVal runControlData As PNS_RUN_CONTROL_DATA) As Integer	
パラメータ	ByVal runControlData As PNS_RUN_CONTROL_DATA	LED ユニットの各色のパターンとブザーを制御する送信データ 詳細は「3.4.1 動作制御データ構造体」を参照
戻り値	Integer	成功：0、失敗：0 以外
説明	PNS コマンドの動作制御コマンドを送信して、LED ユニットの各色のパターンとブザーを制御する	

関数の使用方法	' メイン関数
	Sub Main()
	' LR5-LAN に接続
	Dim ret As Integer
	ret = SocketOpen("192.168.10.1", 10000)
	If ret = -1 Then
	Return
	End If
	
	' PNS コマンドの動作制御コマンド送信
	' LED パターン 0：消灯
	' LED パターン 1：点灯
	' LED パターン 2：点滅(低速)
	' LED パターン 3：点滅(中速)
	' LED パターン 4：点滅(高速)
	' LED パターン 5：シングルフラッシュ
	' LED パターン 6：ダブルフラッシュ
	' LED パターン 7：トリプルフラッシュ
	' LED パターン 9：変化なし
	' ブザーパターン 0：停止
	' ブザーパターン 1：吹鳴
	' ブザーパターン 9：変化なし
	Dim runControlData As PNS_RUN_CONTROL_DATA = New PNS_RUN_CONTROL_DATA With {
	.ledRedPattern = PNS_RUN_CONTROL_LED_ON,
	.ledAmberPattern = PNS_RUN_CONTROL_LED_BLINKING_SLOW,
	.ledGreenPattern = PNS_RUN_CONTROL_LED_NO_CHANGE,
	.ledBluePattern = PNS_RUN_CONTROL_LED_ OFF,
	.ledWhitePattern = PNS_RUN_CONTROL_LED_FLASHING_TRIPLE,
	.buzzerPattern = PNS_RUN_CONTROL_BUZZER_RING
	}
	PNS_RunControlCommand(runControlData)
	
	' ソケットをクローズ
	SocketClose()
	End Sub	

備考	プログラムの概要は「4.4PNS コマンドの動作制御コマンド送信」を参照	

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.2.6. PNS コマンドのクリアコマンド送信

関数名	Public Function PNS_ClearCommand() As Integer	
パラメータ	なし	
戻り値	Integer	成功:0、失敗:0 以外
説明	PNS コマンドのクリアコマンドを送信して、LED ユニットを消灯し、ブザーを停止する	

関数の使用方法	' メイン関数
	Sub Main()
	　' LR5-LAN に接続
	　Dim ret As Integer
	　ret = SocketOpen("192.168.10.1", 10000)
	　If ret = -1 Then
	　　Return
	　End If
	
	　' PNS コマンドのクリアコマンド送信
	　PNS_ClearCommand()
	
	　' ソケットをクローズ
	　SocketClose()
	End Sub	

備考	プログラムの概要は「4.5PNS コマンドのクリアコマンド送信」を参照	

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.2.7. PNS コマンドの状態取得コマンド送信

関数名	Public Function PNS_GetDataCommand(ByRef statusData As PNS_STATUS_DATA) As Integer	
パラメータ	ByRef statusData As PNS_STATUS_DATA	状態取得コマンドの受信データ(LED ユニットおよびブザーの状態) 詳細は「3.4.2 動作制御の状態データ」を参照
戻り値	Integer	成功:0、失敗:0 以外
説明	PNS コマンドの状態取得コマンドを送信して、LED ユニットおよびブザーの状態を取得する	

関数の使用方法	' メイン関数
	Sub Main()
	' LR5-LAN に接続
	Dim ret As Integer
	ret = SocketOpen("192.168.10.1", 10000)
	If ret = -1 Then
	Return
	End If
	
	' PNS コマンドの状態取得コマンド送信
	Dim statusData As PNS_STATUS_DATA = New PNS_STATUS_DATA
	PNS_GetDataCommand(statusData)
	
	' ソケットをクローズ
	SocketClose()
	End Sub	

備考	プログラムの概要は「4.6PNS コマンドの状態取得コマンド送信」を参照	

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.3. 定数説明

3.3.1. 製品区分

定数名	値	説明
PNS_PRODUCT_ID	0x4142	LR5-LAN の製品区分

3.3.2. PNS コマンド識別子

PNS_RUN_CONTROL_COMMAND	0x53	動作制御コマンド
PNS_CLEAR_COMMAND	0x43	クリアコマンド
PNS_GET_DATA_COMMAND	0x47	状態取得コマンド

3.3.3. PNS コマンドの応答データ

定数名	値	説明
PNS_ACK	0x06	正常応答
PNS_NAK	0x15	異常応答

3.3.4. 動作制御コマンドの LED ユニットパターン

定数名	値	説明
PNS_RUN_CONTROL_LED_ON	0x00	消灯
PNS_RUN_CONTROL_LED_OFF	0x01	点灯
PNS_RUN_CONTROL_LED_BLINKING_SLOW	0x02	点滅（低速）
PNS_RUN_CONTROL_LED_BLINKING_MEDIUM	0x03	点滅（低速）
PNS_RUN_CONTROL_LED_BLINKING_HIGH	0x04	点滅（低速）
PNS_RUN_CONTROL_LED_FLASHING_SINGLE	0x05	シングルフラッシュ
PNS_RUN_CONTROL_LED_FLASHING_DOUBLE	0x06	ダブルフラッシュ
PNS_RUN_CONTROL_LED_FLASHING_TRIPLE	0x07	トリプルフラッシュ
PNS_RUN_CONTROL_LED_NO_CHANGE	0x09	変化なし

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.3.5. 動作制御コマンドのブザーパターン

定数名	値	説明
PNS_RUN_CONTROL_BUZZER_STOP	0x00	停止
PNS_RUN_CONTROL_BUZZER_RING	0x01	吹鳴
PNS_RUN_CONTROL_BUZZER_NO_CHANGE	0x09	変化なし

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

3.4. 構造体説明

3.4.1. 動作制御データ構造体

名前	PNS_RUN_CONTROL_DATA
定義	Public Class PNS_RUN_CONTROL_DATA ' LED ユニット赤色のパターン Public ledRedPattern As Byte = 0 ' LED ユニット黄色のパターン Public ledAmberPattern As Byte = 0 ' LED ユニット緑色のパターン Public ledGreenPattern As Byte = 0 ' LED ユニット青色のパターン Public ledBluePattern As Byte = 0 ' LED ユニット白色のパターン Public ledWhitePattern As Byte = 0 ' ブザーの状態 Public buzzerMode As Byte = 0 End Class
説明	動作制御コマンドで送信するデータエリアの LED ユニットの各色のパターンとブザー状態の構造体

3.4.2. 動作制御の状態データ

名前	PNS_STATUS_DATA
定義	Public Class PNS_STATUS_DATA ' LED パターン 1～5 Public ledPattern As Byte() = New Byte(5) {} ' ブザーモード Public buzzer As Byte = 0 End Class
説明	動作制御の状態取得コマンドの応答データの LED ユニットおよびブザーの状態の構造体

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

4. プログラム概要

プログラムの動作を要点のみ記載する。

4.1. LR5-LAN に接続

プログラム	説明
Main.vb `Private sock As Socket = Nothing`	→ソケットのメンバ変数を定義

Main.vb SocketOpen()
 	
`Public Function SocketOpen(ByVal ip As String, ByVal port`
`    Try`
`        ' Set the IP address and port`
`        Dim ipAddress As IPAddress = IPAddress.Parse(ip)`
`        Dim remoteEP As IPEndPoint = New IPEndPoint(ipAdc`
`        sock = New Socket(ipAddress.AddressFamily, Socket`
 	
`        ' Create a socket`
`        If sock Is Nothing Then`
`            Console.WriteLine("failed to create socket")`
`            Return -1`
`        End If`
 	
`        ' Connect to LA-POE`
`        sock.Connect(remoteEP)`
`    Catch ex As Exception`
`        Console.WriteLine(ex.Message)`
`        SocketClose()`
`        Return -1`
`    End Try`
 	
`    Return 0`
`End Function`
	→機器の IP アドレスとポート番号を指定
	デフォルトの IP アドレス：192.168.10.1
	デフォルトのポート番号：10000
	→ソケットのインスタンスを作成
	
	→ソケットの Connect 関数で機器に接続

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

4.2. ソケットをクローズ

プログラム	説明

Main.vb SocketClose()
 	
Public Sub SocketClose()
    If sock IsNot Nothing Then
        ' Close the socket.
        sock.Shutdown(SocketShutdown.Both)
        sock.Close()
    End If
End Sub
	→ソケットをシャットダウンしてからクローズを呼び出す

4.3. コマンドを送信

各コマンドの送信データフォーマットの送信データを作成し、LR5-LAN にコマンドデータを送信する

各コマンドの送信データフォーマットは「4.4PNS コマンドの動作制御コマンド送信」以降を参照

プログラム	説明

Main.vb SendCommand()
 	
If sock Is Nothing Then
    Console.WriteLine("socket is not")
    Return -1
End If
 	
' Send
ret = sock.Send(sendData)
If ret < 0 Then
    Console.WriteLine("failed to send")
    Return -1
End If
 	
' Receive response data
Dim bytes As Byte() = New Byte(1023) {}
Dim recvSize As Integer = sock.Receive(bytes)
If recvSize < 0 Then
    Console.WriteLine("failed to recv")
    Return -1
End If
 	
recvData = New Byte(recvSize - 1) {}
Array.Copy(bytes, recvData, recvSize)
	→作成した送信データを Send 関数で送信
	
	→送信後に Recive 関数で機器からのレスポンスを取得

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

4.4. PNS コマンドの動作制御コマンド送信

プログラム	説明

Main.vb PNS_RunControlCommand()
`Dim sendData As Byte() = {}`
 	
`' 製品区分(AB)`
`sendData = sendData.Concat(BitConverter.GetBytes(PNS_PROD`
 	
`' コマンド識別子(S)`
`sendData = sendData.Concat(New Byte() {PNS_RUN_CONTROL_CC`
 	
`' 空き(0)`
`sendData = sendData.Concat(New Byte() {0}).ToArray()`
 	
`' データサイズ、データエリア`
`Dim data As Byte() = {`
`  runControlData.ledRedPattern,  ' LEDユニット赤色の`
`  runControlData.ledAmberPattern,  ' LEDユニット黄色`
`  runControlData.ledGreenPattern,  ' LEDユニット緑色`
`  runControlData.ledBluePattern,  ' LEDユニット青色(`
`  runControlData.ledWhitePattern,  ' LEDユニット白色`
`  runControlData.buzzerMode  ' ブザーの状態`
`}`
`sendData = sendData.Concat(BitConverter.GetBytes(CUShort(`
`sendData = sendData.Concat(data).ToArray()`
 	
`' PNSコマンドを送信`
`Dim recvData As Byte() = {}`
`ret = SendCommand(sendData, recvData)`
`If ret <> 0 Then`
`  Console.WriteLine("failed to send data")`
`  Return -1`
`End If`
 	
`' 応答データを確認`
`If recvData(0) = PNS_NAK Then`
`  ' 異常応答を受信`
`  Console.WriteLine("negative acknowledge")`
`  Return -1`
`End If`
	以下の順で送信データを作成
	→1 バイト目：製品区分（A：0x41）
	→2 バイト目：製品区分（B：0x42）
	→3 バイト目：識別子（S：0x53）
	→4 バイト目：空き（0x00）
	→5 バイト目：データサイズ（0x00）
	→6 バイト目：データサイズ（0x06）
	→7～12 バイト目：データエリア
	データサイズは 6 バイト
	データエリアには「3.4.1 動作制御データ構造体」の値を設定する
	
	→「4.3 コマンドを送信・受信」を呼び出し、機器にデータを送信
	
	→送信後に応答データを確認
	正常応答：ACK（0x06）
	異常応答：NAK（0x15）

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

4.5. PNS コマンドのクリアコマンド送信

プログラム	説明

Main.vb PNS_ClearCommand()
 	
Dim sendData As Byte() = {}
 	
' Product Category (AB)
sendData = sendData.Concat(BitConverter.GetBytes(PNS_PRODI
 	
' Command identifier (C)
sendData = sendData.Concat(New Byte() {PNS_CLEAR_COMMAND}
 	
' Empty (0)
sendData = sendData.Concat(New Byte() {0}).ToArray()
 	
' Data size
sendData = sendData.Concat(BitConverter.GetBytes(CUShort(I
 	
' Send PNS command
Dim recvData As Byte() = {}
ret = SendCommand(sendData, recvData)
If ret <> 0 Then
Console.WriteLine("failed to send data")
Return -1
End If
 	
' check the response data
If recvData(0) = PNS_NAK Then
' receive abnormal response
Console.WriteLine("negative acknowledge")
Return -1
End If
	以下の順で送信データを作成
	→1 バイト目：製品区分（A：0x41）
	→2 バイト目：製品区分（B：0x42）
	→3 バイト目：識別子（C：0x43）
	→4 バイト目：空き（0x00）
	→5 バイト目：データサイズ（0x00）
	→6 バイト目：データサイズ（0x00）
	データサイズは 0 バイト
	データエリアは無し
	
	→「4.3 コマンドを送信・受信」を呼び出し、機器にデータを送信
	
	→送信後に応答データを確認
	正常応答：ACK（0x06）
	異常応答：NAK（0x15）

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved

4.6. PNS コマンドの状態取得コマンド送信

プログラム	説明

Main.vb PNS_GetDataCommand()
 	
Dim sendData As Byte() = {}
 	
' Product Category (AB)
sendData = sendData.Concat(BitConverter.GetBytes(PNS_PROD
 	
' Command identifier (G)
sendData = sendData.Concat(New Byte() {PNS_GET_DATA_COMMA
 	
' Empty (0)
sendData = sendData.Concat(New Byte() {0}).ToArray()
 	
' Data size
sendData = sendData.Concat(BitConverter.GetBytes(CShort(0
 	
' Send PNS command
Dim recvData As Byte() = {}
ret = SendCommand(sendData, recvData)
If ret <> 0 Then
Console.WriteLine("failed to send data")
Return -1
End If
 	
' check the response data
If recvData(0) = PNS_NAK Then
' receive abnormal response
Console.WriteLine("negative acknowledge")
Return -1
End If
 	
' LED Pattern 1 to 5
statusData.ledPattern = New Byte(5) {}
Array.Copy(recvData, statusData.ledPattern, statusData.le
 	
' Buzzer Mode
statusData.buzzer = recvData(5)
	以下の順で送信データを作成
	→1 バイト目：製品区分（A：0x41）
	→2 バイト目：製品区分（B：0x42）
	→3 バイト目：識別子（G：0x47）
	→4 バイト目：空き（0x00）
	→5 バイト目：データサイズ（0x00）
	→6 バイト目：データサイズ（0x00）
	データサイズは 0 バイト
	データエリアは無し
	
	→「4.3 コマンドを送信・受信」を呼び出し、機器にデータを送信
	
	→送信後に応答データを確認
	正常応答：「3.4.2 動作制御の状態データ」の応答データが取得される
	異常応答：NAK（0x15）
	
	以下の処理で応答データの各データの取得
	→1～5 バイト目：LED ユニットの状態
	・1 バイト目：LED ユニット赤色の状態
	・2 バイト目：LED ユニット黄色の状態
	・3 バイト目：LED ユニット緑色の状態
	・4 バイト目：LED ユニット青色の状態
	・5 バイト目：LED ユニット白色の状態
	・6 バイト目：ブザーの状態

CONFIDENTIAL

PATLITE Corporation 2025
All rights reserved