AVIRTU Controller Integration Guide (C#, VB.NET SDK)

SDK Version: 1.7

Compatible Platforms: .NET Framework 4.x, .NET Core, .NET 6+

Dependencies: System.IO.Ports, System.Threading.Tasks

1). Introduction

This document outlines how to integrate the AVIRTU IO Board into a C#,VB.NET application using the AVIRTUClient class. The SDK utilizes the Modbus RTU protocol over a USB-RS485 connection.

Key Features:

- Thread-Safe: Built-in locking mechanisms for serial port access.
- Robust Output Control: Includes automatic verification (read-after-write) and retry logic.
- Real-time Monitoring: Efficient input polling with busy-state protection.
- Relation Management: Ability to configure or cancel onboard IO relations (e.g., Interlocking).

---

2). Code Example for main functions

1.Initialization & Connection

To begin, instantiate the AVIRTUClient class. You must configure the serial port settings and timing parameters to match your hardware environment.

Code Example

For C#.NET

```
using System;
using System.Threading.Tasks;
using System.Windows.Forms;
using AVIRTU;


namespace MyApp
{
    public partial class MainForm : Form
    {
        private AVIRTUClient _io;
```

```
public MainForm()
{
    InitializeComponent();
}

// ========================================
// 1. เชื่อมต่อระบบ
// ========================================
private void MainForm_Load(object sender, EventArgs e)
{
    try
    {
        _io = new AVIRTUClient("COM1", 9600)
        {
            SlaveId = 1,
            DebounceLimit = 3,
            AutoPollIntervalMs = 50
        };

        // รับ Event ทั้งของเข้าและของออก มาประเมินผลที่เดียวกัน
        _io.OnInputRisingEdge += Io_OnInputChanged;
        _io.OnInputFallingEdge += Io_OnInputChanged;

        _io.StartAutoPolling();

        MessageBox.Show("เชื่อมต่อบอร์ด IO สำเร็จ!");
    }
    catch (Exception ex)
    {
        MessageBox.Show($"เชื่อมต่อล้มเหลว: {ex.Message}");
    }
}
```

```
private void Io_OnInputChanged(object sender, IoEventArgs e)
{
    EvaluateSystemState();
}

// ========================================
// 2. ศูนย์รวม Logic (State-Machine แบบเปิดกว้าง)
// ========================================
private void EvaluateSystemState()
{
    if (this.InvokeRequired)
    {
        this.BeginInvoke(new Action(EvaluateSystemState));
        return;
    }

    // ดึงสถานะปัจจุบันทั้งหมด 8 ช่อง
    bool[] inputs = _io.ReadInputs();

    // ------------------------------------------------------
    // ปรับแต่ง Logic การทำงานของระบบคุณได้ที่นี่ (ตัวอย่างการใช้งาน)
    // ------------------------------------------------------

    if (inputs[0] && inputs[1])
    {
        // เงื่อนไข: เซนเซอร์ช่อง 0 และ 1 ต้องทำงานพร้อมกัน
        lblStatus.Text = "สถานะ: เงื่อนไขครบถ้วน เริ่มกระบวนการทำงานได้!";

        // ตัวอย่าง: สั่งเปิด Relay ช่อง 2
        _ = Task.Run(() => SendOutputCmd(2, true));
    }
    else if (inputs[0] && !inputs[1])
    {
```

```
            // เงื่อนไข: มาแค่ตัวแรก ตัวที่สองยังไม่มา
            lblStatus.Text = "สถานะ: เซนเซอร์ตัวแรกทำงานแล้ว รอตัวถัดไป...";
        }
        else
        {
            // สถานะว่าง หรือเงื่อนไขไม่ตรงตามที่กำหนด
            lblStatus.Text = "สถานะ: รอรับสัญญาณ / ระบบพร้อมทำงาน...";

            // สั่งปิด Relay คืนสถานะเดิม
            _ = Task.Run(() => SendOutputCmd(2, false));
        }
    }

    // ==========================================
    // 3. ฟังก์ชันสั่ง Output ปลอดภัยไม่ค้าง
    // ==========================================
    private bool SendOutputCmd(int channel, bool turnOn)
    {
        if (_io == null) return false;
        OutputAction action = turnOn ? OutputAction.Open : OutputAction.Close;
        return _io.WriteOutput(channel, action);
    }

    private void MainForm_FormClosing(object sender, FormClosingEventArgs e)
    {
        _io?.Dispose();
    }
  }
}
```

For VB.NET

```
Imports System.Threading.Tasks
Imports AVIRTU

Public Class MainForm

    Private WithEvents _io As AVIRTUClient

    ' ========================================
    ' 1. เชื่อมต่อระบบ
    ' ========================================
    Private Sub MainForm_Load(sender As Object, e As EventArgs) Handles
MyBase.Load
        Try
            _io = New AVIRTUClient("COM1", 9600) With {
                .SlaveId = 1,
                .DebounceLimit = 3,
                .AutoPollIntervalMs = 50
            }
            _io.StartAutoPolling()
            MessageBox.Show("เชื่อมต่อบอร์ด IO สำเร็จ!")
        Catch ex As Exception
            MessageBox.Show($"เชื่อมต่อล้มเหลว: {ex.Message}")
        End Try
    End Sub

    ' รับ Event ทั้งขาขึ้นและขาลง มาเข้าที่เดียวกัน
    Private Sub Io_OnInputRisingEdge(sender As Object, e As IoEventArgs) Handles
_io.OnInputRisingEdge
        EvaluateSystemState()
    End Sub
```

```
    Private Sub Io_OnInputFallingEdge(sender As Object, e As IoEventArgs) Handles
_io.OnInputFallingEdge
        EvaluateSystemState()
    End Sub

    ' ========================================
    ' 2. ศูนย์รวม Logic (State-Machine แบบเปิดกว้าง)
    ' ========================================
    Private Sub EvaluateSystemState()
        Me.BeginInvoke(Sub()
                           ' ดึงสถานะปัจจุบันทั้งหมด 8 ช่อง
                           Dim inputs As Boolean() = _io.ReadInputs()

                           ' ----------------------------------------------------
                           ' ปรับแต่ง Logic การทำงานของระบบคุณได้ที่นี่ (ตัวอย่างการใช้งาน)
                           ' ----------------------------------------------------

                           If inputs(0) AndAlso inputs(1) Then
                               ' เงื่อนไข: เซนเซอร์ช่อง 0 และ 1 ต้องทำงานพร้อมกัน
                               lblStatus.Text = "สถานะ: เงื่อนไขครบถ้วน เริ่มกระบวนการทำงานได้!"

                               ' ตัวอย่าง: สั่งเปิด Relay ช่อง 2
                               Task.Run(Function() SendOutputCmd(2, True))

                           ElseIf inputs(0) AndAlso Not inputs(1) Then
                               ' เงื่อนไข: มาแค่ตัวแรก ตัวที่สองยังไม่มา
                               lblStatus.Text = "สถานะ: เซนเซอร์ตัวแรกทำงานแล้ว รอตัวถัดไป..."
```

```
                Else
                    ' สถานะว่าง หรือเงื่อนไขไม่ตรงตามที่กำหนด
                    lblStatus.Text = "สถานะ: รอรับสัญญาณ / ระบบพร้อมทำงาน..."

                    ' สั่งปิด Relay คืนสถานะเดิม
                    Task.Run(Function() SendOutputCmd(2, False))
                End If
            End Sub)
    End Sub

    ' ========================================
    ' 3. ฟังก์ชันสั่ง Output ปลอดภัยไม่ค้าง
    ' ========================================
    Private Function SendOutputCmd(channel As Integer, turnOn As Boolean) As
Boolean
        If _io Is Nothing Then Return False
        Dim action As OutputAction = If(turnOn, OutputAction.Open, OutputAction.Close)
        Return _io.WriteOutput(channel, action)
    End Function

    Private Sub MainForm_FormClosing(sender As Object, e As FormClosingEventArgs)
Handles Me.FormClosing
        If _io IsNot Nothing Then
            _io.Dispose()
        End If
    End Sub

End Class
```

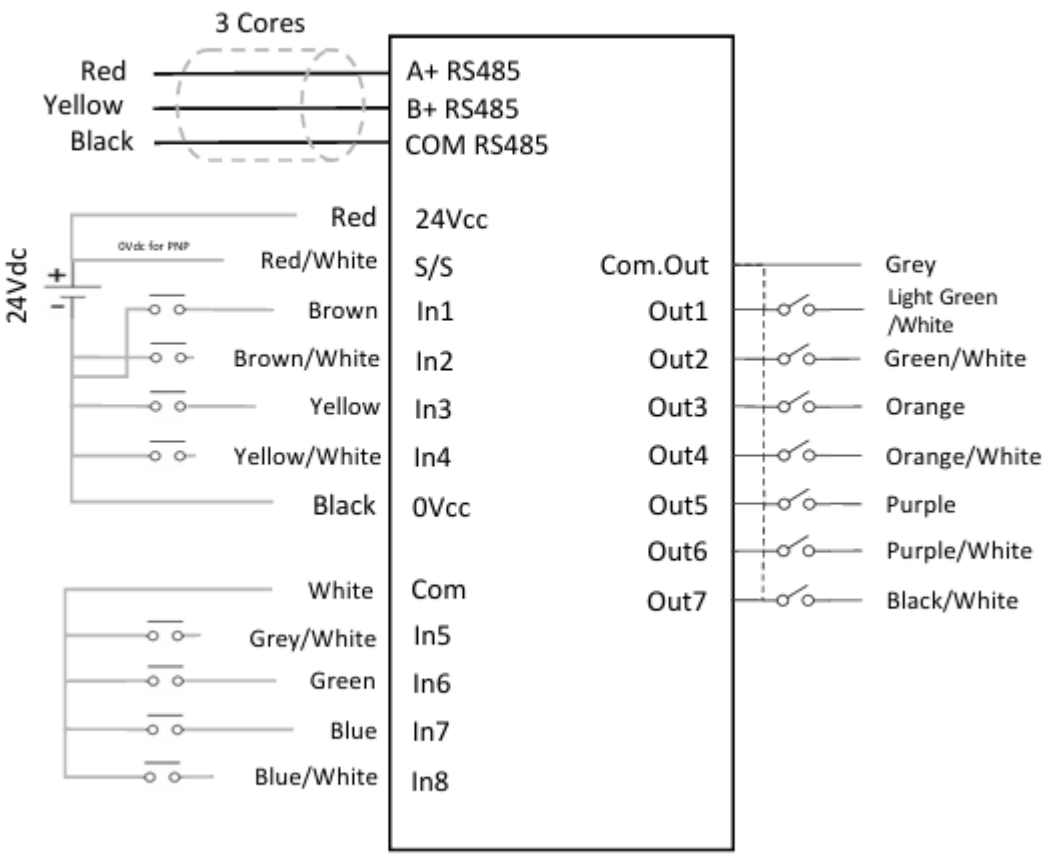
3). Wiring diagrams

Digital I/O module : AVI24-447

Group	Wire Color	Signal	Description
3-Cores Cable			
	Red	A+	RS485 Data + [1]
	Yellow	B-	RS485 Data - [2]
	Black	GND	RS485 Signal Ground [3]
20-Cores Cable			
Power	Red	24Vcc	Power Supply +24VDC [4]
	Black	0Vcc	Power Supply 0V (GND) [5]
Input (Wet)	Red/White	S/S (Wet)	Common (Sink/Source) for Input 1-4 [6]
	Brown	In 1 (Wet)	Digital Input 1 [7]
	Brown/White	In 2 (Wet)	Digital Input 2 [8]
	Yellow	In 3 (Wet)	Digital Input 3 [9]
	Yellow/White	In 4 (Wet)	Digital Input 4 [10]
Input (Dry)	White	COM In (Dry)	Common for Input 5-8 [11]
	Grey/White	In 5 (Dry)	Digital Input 5 [12]
	Green	In 6 (Dry)	Digital Input 6 [13]
	Blue	In 7 (Dry)	Digital Input 7 [14]

Group	Wire Color	Signal	Description
	Blue/White	In 8 (Dry)	Digital Input 8 [15]
Output (Dry)	Grey	COM Output	Common for Output 1-7 [16]
	Light Green/White	Out 1	Digital Output 1 [17]
	Green/White	Out 2	Digital Output 2 [18]
	Orange	Out 3	Digital Output 3 [19]
	Orange/White	Out 4	Digital Output 4 [20]
	Purple	Out 5	Digital Output 5 [21]
	Purple/White	Out 6	Digital Output 6 [22]
	Black/White	Out 7	Digital Output 7 [23]

AVIRTU : AVI24-447

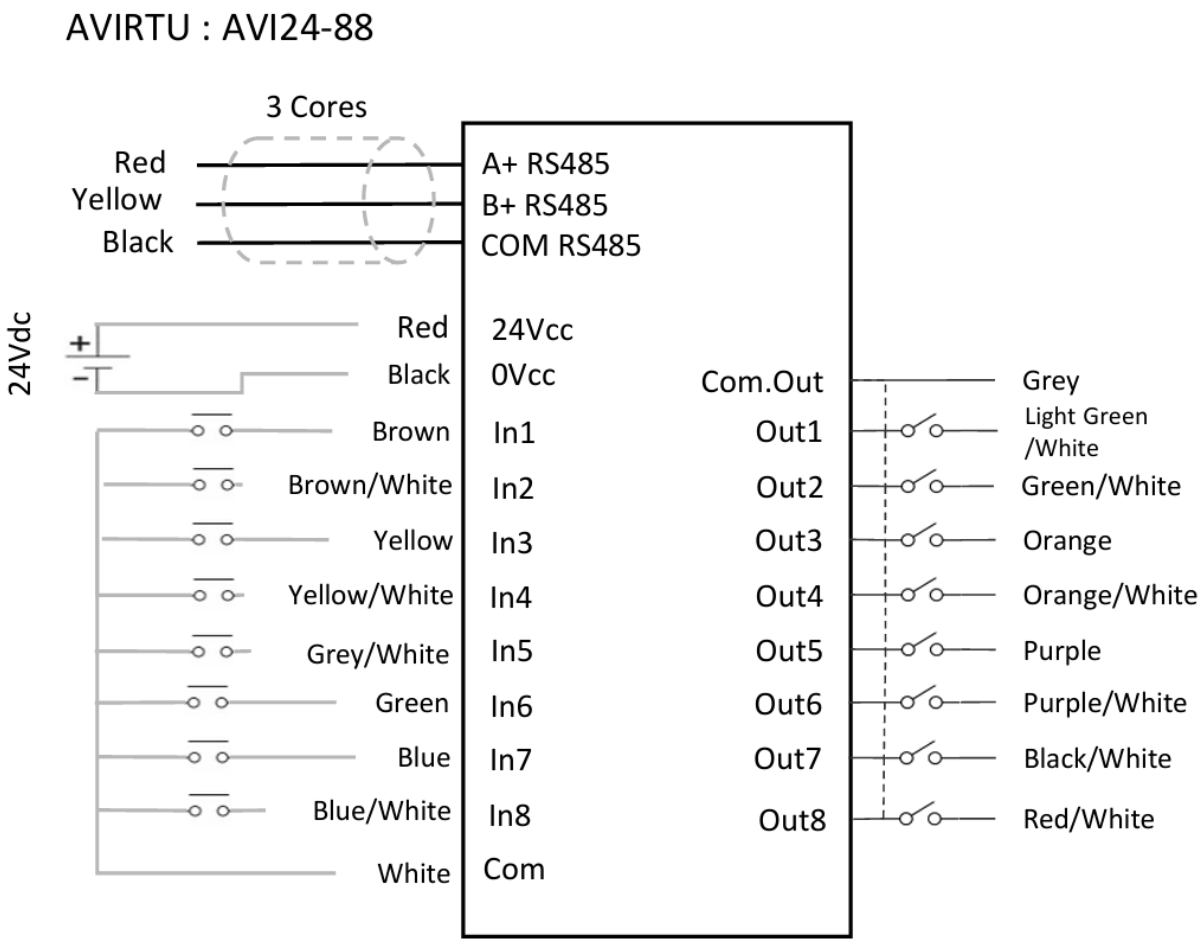


Digital I/O module : AVI24-88

Group	Wire Color	Signal	Description
3-Cores Cable			
	Red	A+	RS485 Data +
	Yellow	B-	RS485 Data -
	Black	GND	RS485 Signal Ground
20-Cores Cable			

Group	Wire Color	Signal	Description
Power & Comm	Red	24Vcc	Power Supply +24VDC
	Black	0Vcc	Power Supply 0V (GND)
Input (Dry)	White	COM In	Common for Input 1-8
	Brown	In 1	Digital Input 1
	Brown/White	In 2	Digital Input 2
	Yellow	In 3	Digital Input 3
	Yellow/White	In 4	Digital Input 4
	Grey/White	In 5	Digital Input 5
	Green	In 6	Digital Input 6
	Blue	In 7	Digital Input 7
	Blue/White	In 8	Digital Input 8
Output (Dry)	Grey	COM Output	Common for Output 1-8
	Light Green/White	Out 1	Digital Output 1
	Green/White	Out 2	Digital Output 2
	Orange	Out 3	Digital Output 3
	Orange/White	Out 4	Digital Output 4
	Purple	Out 5	Digital Output 5

Group	Wire Color	Signal	Description
	Purple/White	Out 6	Digital Output 6
	Black/White	Out 7	Digital Output 7
	Red/White	Out 8	Digital Output 8



4). Direct Serial Control (Modbus RTU, HEX)

If the SDK is not used, you can control the board by sending Modbus RTU commands directly via RS485.

Settings: Baud Rate 9600, Data bits 8, Parity None, Stop bits 1 (9600 8N1), Slave ID = 01, CRC16 Modbus.

2.1 Check Input Status (0–8)

Function 03: Read Input Status (Address 0x0081–0x0088)

Ch	Read Command (HEX + CRC)	Response (ON)	Response (OFF)	Description
1	01 03 00 81 00 01 D4 22	01 03 02 00 01 79 84	01 03 02 00 00 B8 44	Read Input 1 (0x0081) 0x0001=ON, 0x0000=OFF
2	01 03 00 82 00 01 24 22	01 03 02 00 01 79 84	01 03 02 00 00 B8 44	Read Input 2 (0x0082) 0x0001=ON, 0x0000=OFF
3	01 03 00 83 00 01 75 E2	01 03 02 00 01 79 84	01 03 02 00 00 B8 44	Read Input 3 (0x0083) 0x0001=ON, 0x0000=OFF
4	01 03 00 84 00 01 C4 23	01 03 02 00 01 79 84	01 03 02 00 00 B8 44	Read Input 4 (0x0084) 0x0001=ON, 0x0000=OFF
5	01 03 00 85 00 01 95 E3	01 03 02 00 01 79 84	01 03 02 00 00 B8 44	Read Input 5 (0x0085) 0x0001=ON, 0x0000=OFF
6	01 03 00 86 00 01 65 E3	01 03 02 00 01 79 84	01 03 02 00 00 B8 44	Read Input 6 (0x0086) 0x0001=ON, 0x0000=OFF
7	01 03 00 87 00 01 34 23	01 03 02 00 01 79 84	01 03 02 00 00 B8 44	Read Input 7 (0x0087) 0x0001=ON, 0x0000=OFF
8	01 03 00 88 00 01 04 20	01 03 02 00 01 79 84	01 03 02 00 00 B8 44	Read Input 8 (0x0088)

Ch	Read Command (HEX + CRC)	Response (ON)	Response (OFF)	Description
				0x0001=ON, 0x0000=OFF

Read All Inputs (0): 01 03 00 81 00 08 14 24

2.2 Control Output ON/OFF (0–8)

Function 06: Write Relay Value (Address 0x0001–0x0008)

Data: 0x0100 = Open (ON), 0x0200 = Close (OFF)

Ch	ON Command (Open)	OFF Command (Close)	Register
1	01 06 00 01 01 00 D9 9A	01 06 00 01 02 00 D9 6A	0x0001
2	01 06 00 02 01 00 29 9A	01 06 00 02 02 00 29 6A	0x0002
3	01 06 00 03 01 00 78 5A	01 06 00 03 02 00 78 AA	0x0003
4	01 06 00 04 01 00 C9 9B	01 06 00 04 02 00 C9 6B	0x0004
5	01 06 00 05 01 00 98 5B	01 06 00 05 02 00 98 AB	0x0005
6	01 06 00 06 01 00 68 5B	01 06 00 06 02 00 68 AB	0x0006
7	01 06 00 07 01 00 39 9B	01 06 00 07 02 00 39 6B	0x0007
8	01 06 00 08 01 00 09 98	01 06 00 08 02 00 09 68	0x0008

- **Turn ON All: 01 06 00 00 07 00 8B FA**
- **Turn OFF All: 01 06 00 00 08 00 8E 0A**
- ***Note: The board will echo the exact command back to confirm a successful operation.***

2.3 Check Output Status (Readback)

Function 03: Read Relay Status (Address 0x0001–0x0008)

Data: 0x0001 = Open (ON), 0x0000 = Close (OFF)

Ch	Read Command	Response (ON)	Response (OFF)
1	01 03 00 01 00 01 D5 CA	01 03 02 00 01 79 84	01 03 02 00 00 B8 44

Ch	Read Command	Response (ON)	Response (OFF)
2	01 03 00 02 00 01 25 CA	01 03 02 00 01 79 84	01 03 02 00 00 B8 44
3	01 03 00 03 00 01 74 0A	01 03 02 00 01 79 84	01 03 02 00 00 B8 44
4	01 03 00 04 00 01 C5 CB	01 03 02 00 01 79 84	01 03 02 00 00 B8 44
5	01 03 00 05 00 01 94 0B	01 03 02 00 01 79 84	01 03 02 00 00 B8 44
6	01 03 00 06 00 01 64 0B	01 03 02 00 01 79 84	01 03 02 00 00 B8 44
7	01 03 00 07 00 01 35 CB	01 03 02 00 01 79 84	01 03 02 00 00 B8 44
8	01 03 00 08 00 01 05 C8	01 03 02 00 01 79 84	01 03 02 00 00 B8 44

Download Configuration Helper Application via QR code below.

Application

Example code