VB24 - TCP/IP network communications with a Picaxe 28x2

DISCLAIMER

HEREIN "*THE CONTENT*" SHALL REFERE TO ANY AND All VIDEOS, CONTENT, SOFTWARE, CODE, EXAMPLES, CIRCUITS, DOWNLOADS, KNOWHOW, AND/OR INFORMATION.

THE CONTENT PRESENTED BY DR. DAN WEDDING IS FOR ENTERTAINMENT PURPOSES ONLY.

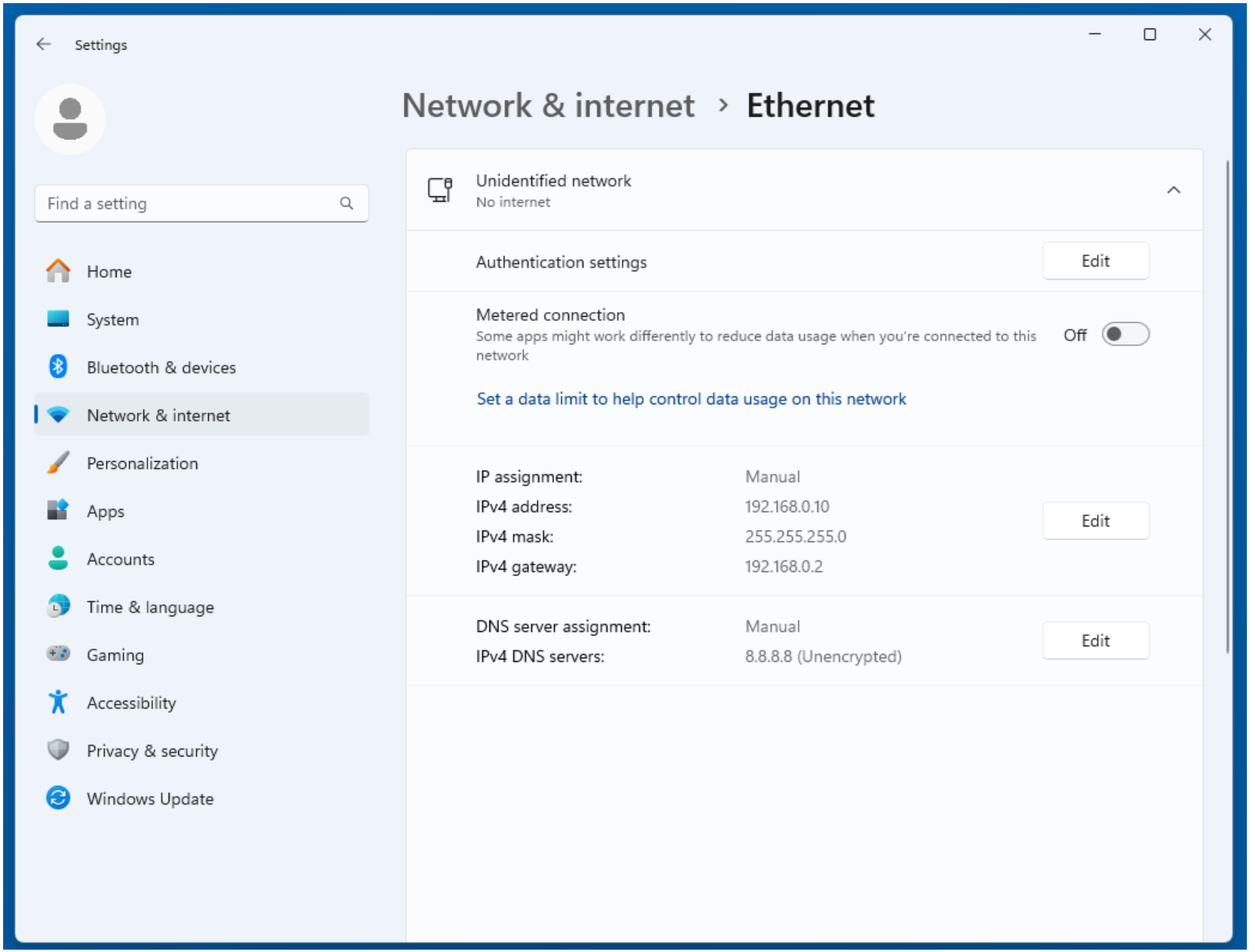
DR. DAN WEDDING MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, AS TO THE ACCURACY OF *THE CONTENT* AND *THE CONTENT* IS PROVIDED "AS-IS".

IN EXCHANGE FOR YOUR USE OF *THE CONTENT*; YOU AGREE INDEMNIFY AND HOLD DR. DAN WEDDING HARMLESS FROM AND AGAINST ANY AND ALL CLAIMS, DAMAGES, COSTS, AND EXPENSES, INCLUDING ATTORNEYS' FEES, ARISING FROM AND/OR RELATED TO THE USE OF *THE CONTENT*.

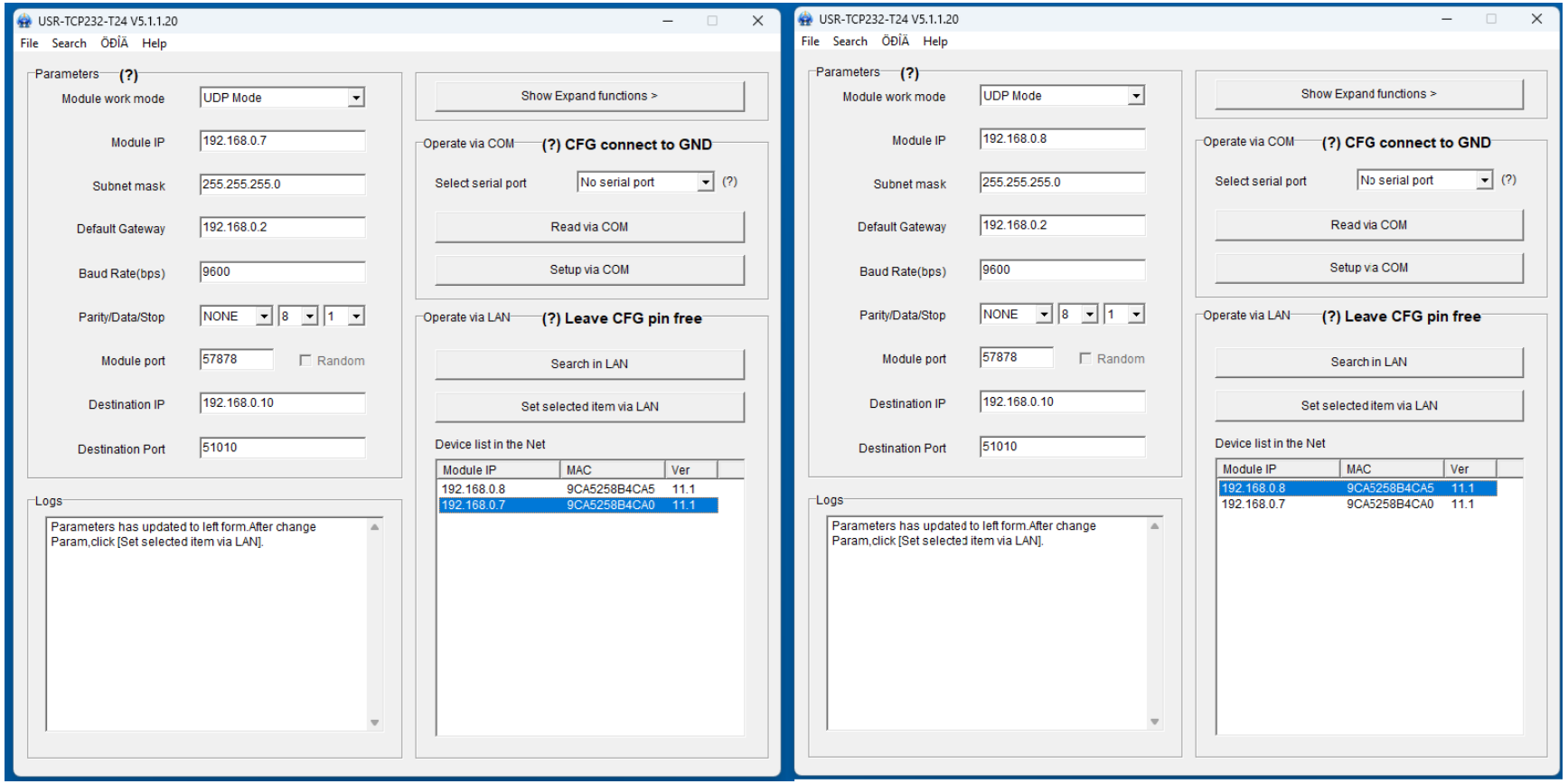
REPRODUCTION, REPUBLICATION, DISTRIBUTION, TRANSMISSION, DISPLAY, OR PERFORMANCE, OF *THE CONTENT* IS PROHIBITED WITHOUT THE EXPRESS WRITTEN CONSENT OF DR. DAN WEDDING.

Dan Wedding PhD

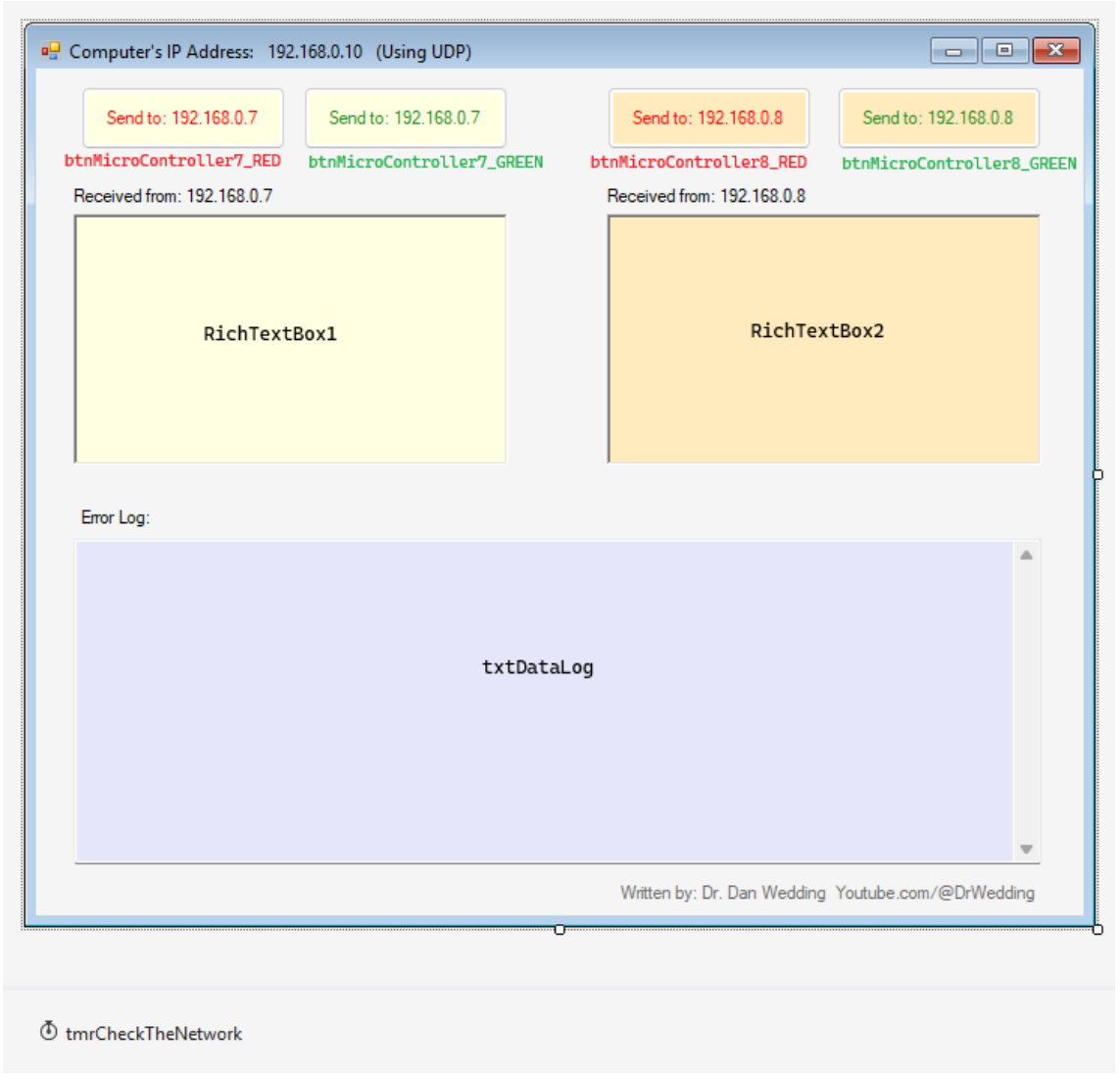
SETTINGS (COMPUTER) - Youtube.com/@DrWedding VB24



SETTINGS (USR-TCP232) - Youtube.com/@DrWedding VB24



VB FORM - Youtube.com/@DrWedding VB24



VB CODE - Youtube.com/@DrWedding VB24

```
''''''''''''''''''''''''''''''''''''''''''''''''''''
'  If you can send info to the uController, but can't get a reply - it might be a firewall setting.
''''''''''''''''''''''''''''''''''''''''''''''''''''

Imports System.Net
Imports System.Text.Encoding


Public Class Form1

    'One socket to write to network - must add ip and port when you write
    'All uControllers are listening to the same port (57878)
    Dim TalkToNetwork As New Sockets.UdpClient(57878)

    'One socket to receive from the network. All uControllers are Talking on port (51010)
    'VB grabs the data as well as the IP the information came from.
    Dim ListenToNetwork As New Sockets.UdpClient(51010)

    'This is a "Blank" IPEndPoint - and endpoint is made up of an IPaddress:Port
    'It gets over written each time data is received on port 51010.
    'All of the microcontrollers are transmitting to 192.168.0.10 on port 51010
    Dim FromIP As IPEndPoint = New IPEndPoint(IPAddress.Any, 51010)


    'This is a byte array to receive the data from the network
    Dim receivebytes() As Byte

    Dim sendbytes() As Byte


    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load

        'Blocking means should the program stop and wait for the data?  If blocking is true - the code will stop
        'wait for the incomming or out going data and all other functions will stop.
        'When Blocking is false the code will check if there is data - if there is no data, it will move on.

        TalkToNetwork.Client.Blocking = False
        ListenToNetwork.Client.Blocking = False

        'The timer will check the incoming data every 100 milliseconds to see if anything arrived.
        tmrCheckTheNetwork.Interval = 100
        tmrCheckTheNetwork.Enabled = True


    End Sub
```

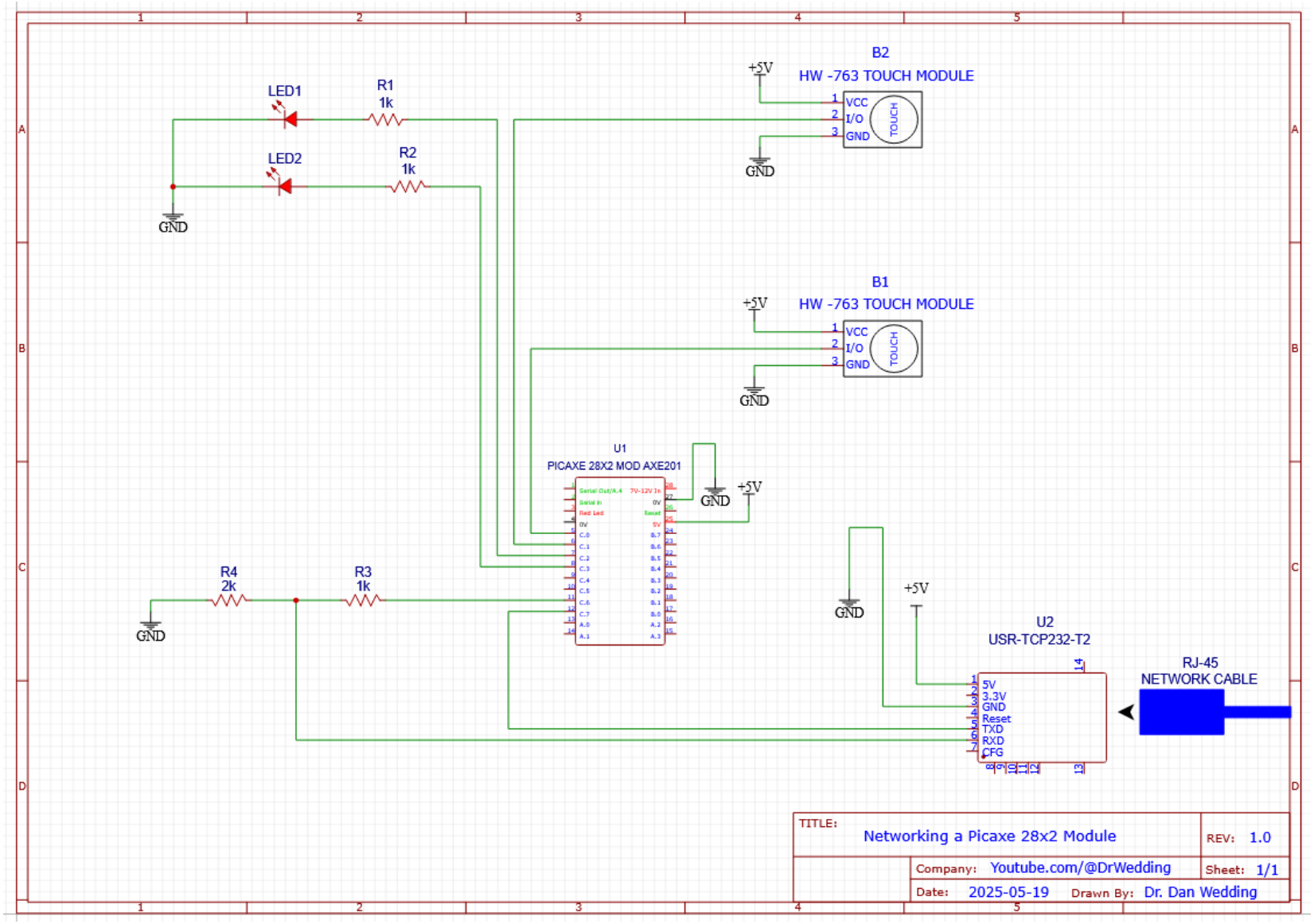
```
Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing
    '
    'Release the resources when you are done
    '
    TalkToNetwork.Dispose()
    ListenToNetwork.Dispose()
End Sub


Private Sub btnMicroController7_RED_Click(sender As Object, e As EventArgs) Handles btnMicroController7_RED.Click

    'Subroutine to toggle the background color of button 1 (192.168.0.7 - RED)
    Toggle_Color(1)

    '
    'RED - Write to 192.168.0.7 on port 57878 when this button is pressed
    '
    TalkToNetwork.Connect("192.168.0.7", 57878)
    sendbytes = ASCII.GetBytes("R")
    TalkToNetwork.Send(sendbytes, sendbytes.Length)

End Sub


Private Sub btnMicroController7_GREEN_Click(sender As Object, e As EventArgs) Handles btnMicroController7_GREEN.Click

    'Subroutine to toggle the background color of button 2 (192.168.0.7 - GREEN)
    Toggle_Color(2)

    '
    ' GREEN- Write to 192.168.0.7 on port 57878 when this button is pressed
    '
    TalkToNetwork.Connect("192.168.0.7", 57878)
    sendbytes = ASCII.GetBytes("G")
    TalkToNetwork.Send(sendbytes, sendbytes.Length)

End Sub
```

```
Private Sub btnMicroController8_RED_Click(sender As Object, e As EventArgs) Handles btnMicroController8_RED.Click

    'Subroutine to toggle the background color of button 3 (192.168.0.8 - RED)
    Toggle_Color(3)

    '
    'Write to 192.168.0.8 on port 57878 when this button is pressed
    '
    TalkToNetwork.Connect("192.168.0.8", 57878)
    sendbytes = ASCII.GetBytes("R")
    TalkToNetwork.Send(sendbytes, sendbytes.Length)

End Sub

Private Sub btnMicroController8_GREEN_Click(sender As Object, e As EventArgs) Handles btnMicroController8_GREEN.Click

    'Subroutine to toggle the background color of button 4 (192.168.0.8 - GREEN)
    Toggle_Color(4)

    '
    'Write to 192.168.0.8 on port 57878 when this button is pressed
    '
    TalkToNetwork.Connect("192.168.0.8", 57878)
    sendbytes = ASCII.GetBytes("G")
    TalkToNetwork.Send(sendbytes, sendbytes.Length)

End Sub
```

```
Private Sub tmrCheckTheNetwork_Tick(sender As Object, e As EventArgs) Handles tmrCheckTheNetwork.Tick
    Try
        '
        'This polled routine listens to port 51010,
        'any data received is stored in receivedbytes
        'the IP address of the sender is stored in FromIP - so data can be routed properly.
        '
        'If no data is there - these are not the droids you are looking for, move along, move along.
        '

        If ListenToNetwork.Available > 0 Then

            receivebytes = ListenToNetwork.Receive(FromIP)

            'If there is data - figure out who sent it (based on IP address)

            If (FromIP.Address.ToString = "192.168.0.7") Then

                'Subroutine to toggle the foreground color of RichTextBox for 192.168.0.7
                Toggle_Color(5)

                RichTextBox1.AppendText(ASCII.GetString(receivebytes))

            ElseIf (FromIP.Address.ToString = "192.168.0.8") Then

                'Subroutine to toggle the foreground color of RichTextBox for 192.168.0.8
                Toggle_Color(6)

                RichTextBox2.AppendText(ASCII.GetString(receivebytes))

            End If

        End If


    Catch ex As Exception

        ' If anything goes wrong - add it to the error log textbox
        txtDataLog.AppendText("Network Error from IP: " & FromIP.Address.ToString & ":" & vbCrLf & ex.ToString & vbCrLf & vbCrLf)

    End Try

End Sub
```

```
  Sub Toggle_Color(ByRef btn)

    If btn = 1 Then
      If btnMicroController7_RED.BackColor = Color.FromArgb(255, 255, 255, 225) Then
        btnMicroController7_RED.BackColor = Color.FromArgb(255, 255, 192, 192)
      Else
        btnMicroController7_RED.BackColor = Color.FromArgb(255, 255, 255, 225)
      End If

    ElseIf btn = 2 Then
      If btnMicroController7_GREEN.BackColor = Color.FromArgb(255, 255, 255, 225) Then
        btnMicroController7_GREEN.BackColor = Color.FromArgb(255, 192, 255, 192)
      Else
        btnMicroController7_GREEN.BackColor = Color.FromArgb(255, 255, 255, 225)
      End If

    ElseIf btn = 3 Then
      If btnMicroController8_RED.BackColor = Color.FromArgb(255, 255, 235, 190) Then
        btnMicroController8_RED.BackColor = Color.FromArgb(255, 255, 192, 192)
      Else
        btnMicroController8_RED.BackColor = Color.FromArgb(255, 255, 235, 190)
      End If

    ElseIf btn = 4 Then
      If btnMicroController8_GREEN.BackColor = Color.FromArgb(255, 255, 235, 190) Then
        btnMicroController8_GREEN.BackColor = Color.FromArgb(255, 192, 255, 192)
      Else
        btnMicroController8_GREEN.BackColor = Color.FromArgb(255, 255, 235, 190)
      End If

    ElseIf btn = 5 Then
      If ASCII.GetString(receivebytes) = "B1 " Then
        RichTextBox1.SelectionColor = Color.Red
      Else
        RichTextBox1.SelectionColor = Color.Blue
      End If

    ElseIf btn = 6 Then
      If ASCII.GetString(receivebytes) = "B1 " Then
        RichTextBox2.SelectionColor = Color.Red
      Else
        RichTextBox2.SelectionColor = Color.Blue
      End If

    End If

  End Sub

End Class
```

PICAXE 28x2 SCHEMATIC - Youtube.com/@DrWedding VB24



PICAXE BOARD PARTS LIST - Youtube.com/@DrWedding VB24

REFERENCE	QUANTITY	DESCRIPTION	PART	NOTE
U1	1	Picaxe 28x2 Module	Picaxe.com Picaxe 28x2 Module	**Note 1**
U2	1	Ethernet to serial UART conversion module	USR-TCP232-T2	**Note 2**
B1-B2	2	Capacitive touch module	TTP223 ALSO CALLED HW-763	
R1-R2	2	LED Current limiting resistor	1kΩ Resistor	**Note 3**
LED1-LED2	2	LED (Colors optional)	Generic LED	
R3	1	First Half of the voltage divider	1kΩ Resistor	**Note 4**
R4	1	Second half of the voltage divider	2kΩ Resistor	**Note 5**

Note 1: A Picaxe 28x2 was used in this video. Any other microcontroller should work, but the code will obviously change.

Note 2: The USR-TCP232-T2 Ethernet to serial UART conversion module does most of the work on the circuit side. This part needs to have its properties configured. The software to configure the part is named: **USR-TCP232-T24-V5.1.1.20.exe** and I downloaded it from the manufacturer's website here: https://www.pusr.com/support/downloads/Setup-Software-USR-TCP232-T24-V51120.html

Note 3: The current limiting resistors for the LEDs in this video were 1kΩ. The LEDs in this video were high efficiency, super bright, and low power. I also had the LEDs aimed directly at the camera. Your resistor values may be different depending on your needs. A lower value resistor, such as a 680Ω, 560Ω, 470Ω, or 330Ω will increase the brightness of the LED significantly.

Note 4: The RXD and TXD pins on the USR-TCP232-T2 can only handle 3.3V. See the bottom of page 14 of the user manual found here: https://www.pusr.com/download/M0/USR-TCP232-T2-User-Manual-V1.1.pdf The 3.3V TXD pin feeding the 5V RDX Arduino pin is fine. This can be a direct connection. However, the 5V TXD feeding the 3.3V RXD pin on the USR-TCP232 can damage the module. There are specialized modules called voltage levelers that do this. I chose to use two resistors to create a Voltage Divider Circuit. The equation is shown below:

$$5V * \frac{R4}{(R3 + R4)} = 5V * \frac{2k}{(1k + 2k)} = 5V * \frac{2k}{3k} = 3.3V$$

By choosing R4 to be 2kΩ and R3 to be 1kΩ, the voltage is exactly 3.3V.

Note 5: If you do not have a 2kΩ resistor, you can make one out of two 1kΩ resistors in series. You can use a potentiometer set to 2kΩ. Some people use a 2.2kΩ resistor in place of the 2kΩ. This will increase the voltage feeding the USR-TCP232-T2 to 3.44V and the maximum voltage the pin should be supplied is 3.45V (*See page 9 of the user manual*).

PICAXE CODE - Youtube.com/@DrWedding VB24

NOTE: Your code may be different if you use a different Picaxe.

```
;RUN ONCE CODE GOES HERE
#no_table
#no_data

symbol once0 = b1
once0 = 1

symbol once1 = b2
once1 = 1

'Button Pins
input c.0, c.1

'LED Pins
output c.2, c.3
low c.2
low c.3

'Serial Pins
output c.6
input c.7
high c.6

hsersetup B9600_8, $9 ;This sets the mode bits (most times it is $9 or $f)
hserptr = 0
hserinflag = 0
;Received values will be moved into B10, so clear it out
b10 = 0x00
```

```
main:

	'Check to see if the first button is being pressed
	if pinc.0 = 1  and once0 = 1 then
		serout c.6, t9600_8, ("B1 ")
		once0 = 0
	elseif pinc.0 = 0  and once0 = 0 then
	once0 = 1
	end if



	'Check to see if the second button is being pressed
	if pinc.1 = 1  and once1 = 1 then
		serout c.6, t9600_8, ("B2 ")
		once1 = 0
	elseif pinc.1 = 0  and once1 = 0 then
	once1 = 1
	end if



	'Check to see if a value showed up in the UART while
	'the uController is doing something else
	'The incoming information was stored in the scratchpad
	if hserinflag != 0 then

		pause 10 ;In case it has not finished reading yet
		get 0, b10

		' Toggle the selected LED
		if b10 = "R" then
			'if I got an R from VB - toggle c.2
			toggle c.2
		else if b10 = "G" then
			'if I got a B from VB - toggle c.3
			toggle c.3
		end if

		'Reset the UART values
		hserptr = 0
		hserinflag = 0
		b10 = 0x00

	end if

goto main
```