VB25 - TCP/IP network communications with an Arduino Metro Mini

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.

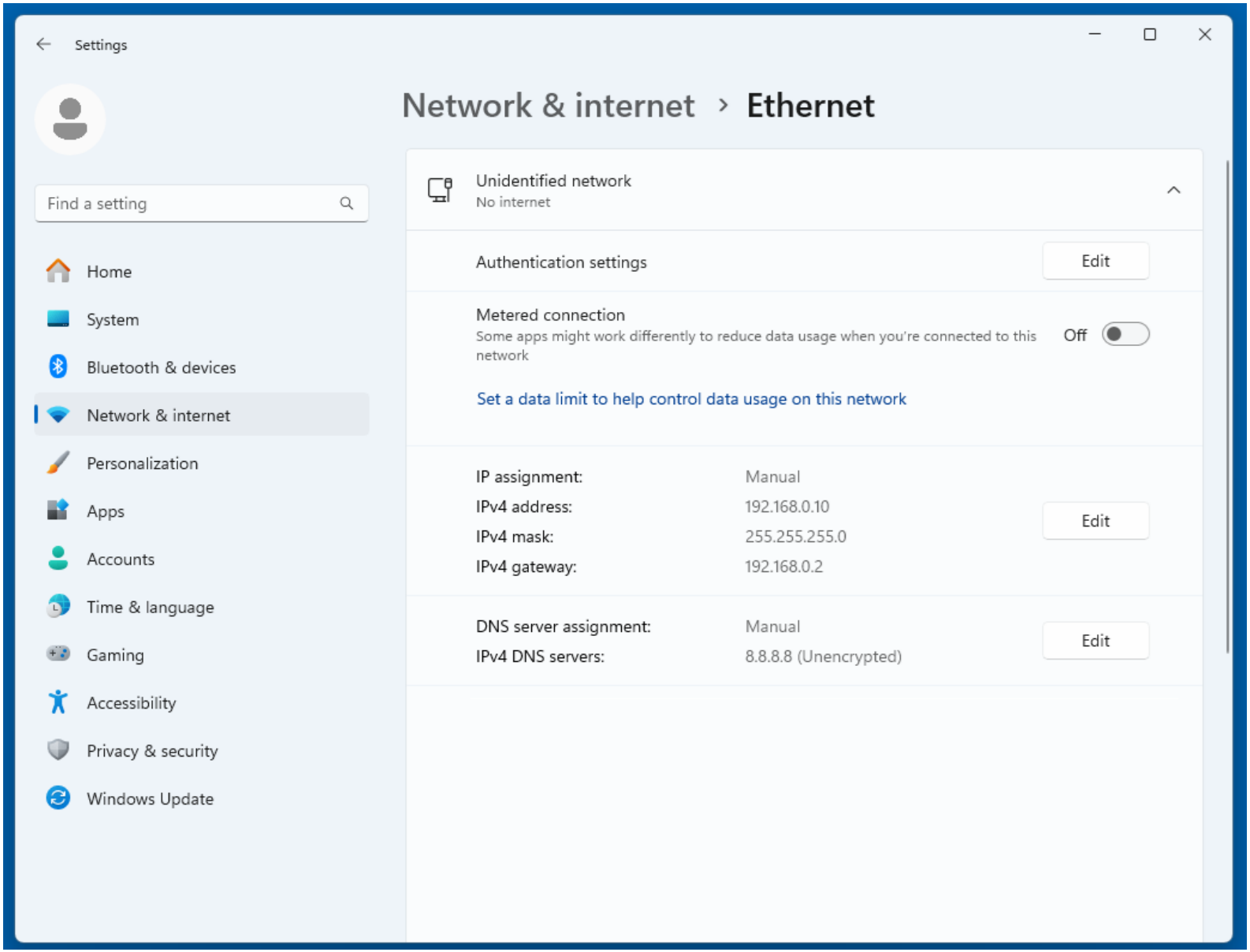
DR. DAN WEDDING MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, AS TO THE ACCURACY OF *THE CONTENT* AND *THE CONTENT* IS PROVIDED "AS-IS".

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*.

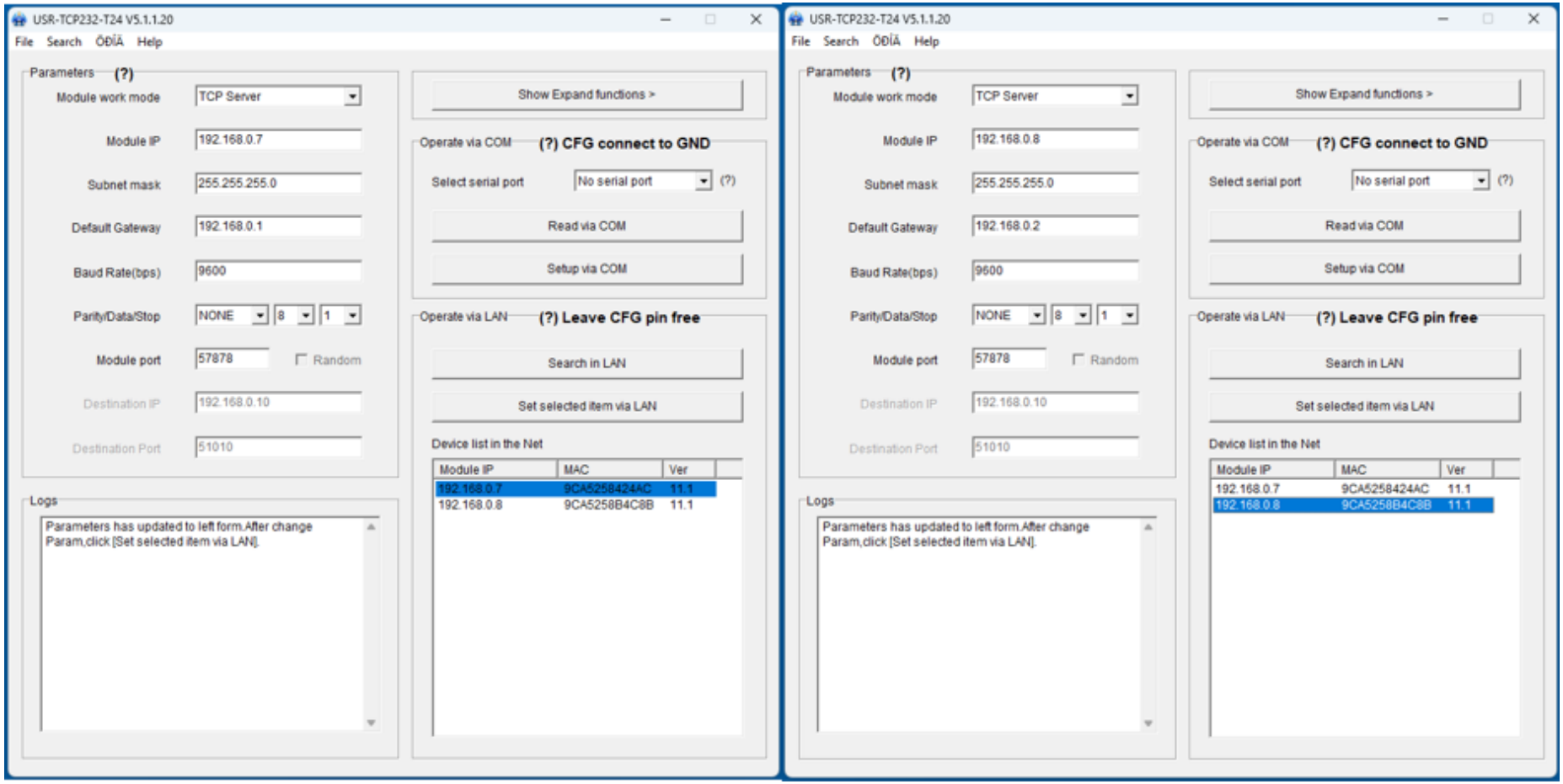
REPRODUCTION, REPUBLICATION, DISTRIBUTION, TRANSMISSION, DISPLAY, OR PERFORMANCE, OF *THE CONTENT* IS PROHIBITED WITHOUT THE EXPRESS WRITTEN CONSENT OF DR. DAN WEDDING.

Dan Wedding PhD

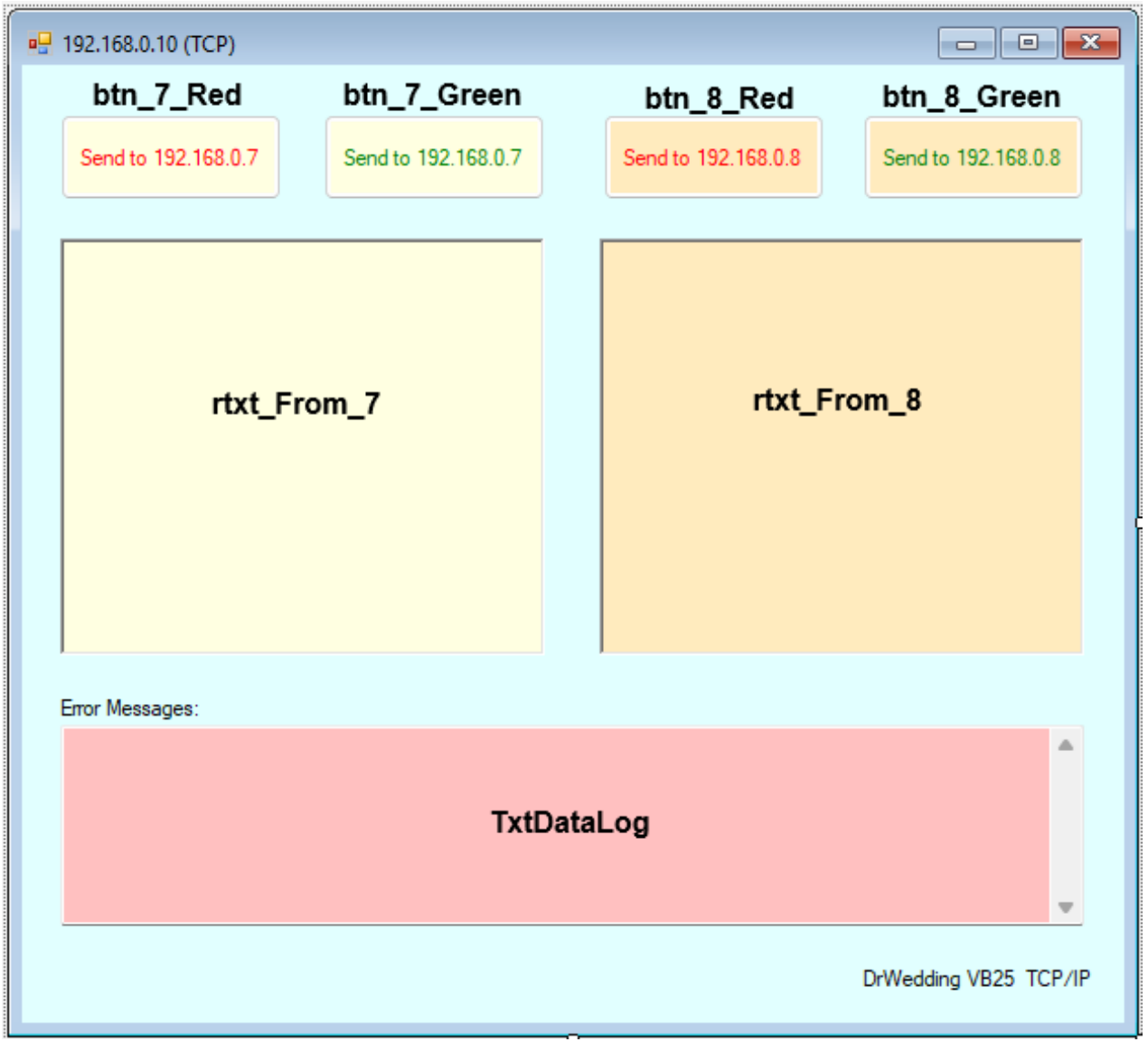
SETTINGS (COMPUTER) - Youtube.com/@DrWedding VB25



SETTINGS (USR-TCP232) - Youtube.com/@DrWedding VB25



VB FORM - Youtube.com/@DrWedding VB25



VB CODE - Youtube.com/@DrWedding VB25

```
'
'Written by: Dan Wedding  Youtube.com/@DrWedding
'
'Date: 2-23-2233
'
'Description:
'
'This code sets up TCP/IP communication with a socket between the uControllers and the computer.
'The uControllers and Computer must be on the same subnet.
'
'   The computer is a client
'   The computer's IP address is 192.168.0.10
'   The computer will accept any port (in case the uControllers are transmitting on different ports)
'   The computer is transmitting on 57878
'
'   The uController007 's IP address is 192.168.0.7
'   The uController007 is set up as a TCP SERVER
'   The uController007 is monitoring port number 57878 for incoming data
'   Inside the uController007 it knows to transmit to the TCP CLIENT at IP address 192.168.0.10
'   Inside the uController007 it knows to transmit on 51010 but it doesn't matter. The computer will look at
'      all of the incoming ports from the uController.
'
'   The uController008 's IP address is 192.168.0.8
'   The uController008 is set up as a TCP SERVER
'   The uController008 is monitoring port number 57878 for incoming data
'   Inside the uController008 it knows to transmit to the TCP CLIENT at IP address 192.168.0.10
'   Inside the uController008 it knows to transmit on 51010 but it doesn't matter. The computer will look at
'      all of the incoming ports from the uController.
'


Imports System

Imports System.Net

Imports System.Net.Sockets

Imports System.Text

Imports System.Threading
```

```
Public Class Form1
    ' GLOBAL VARIABLES

    'If you need the computer's IP address, this is the command. But it is not used in this code.
    'Public ComputerIPAddress As IPHostEntry = Dns.GetHostEntry(Dns.GetHostName())

    ' ***** uController007 information:

    'Load the IP address of the uController into uControllerIPAddress
    Dim uController007IPAddress As IPAddress = IPAddress.Parse("192.168.0.7")

    'Combine the IP Adress of the uController and the Port Number into one IPEndPoint variable
    Dim uController007IPandPort As New IPEndPoint(uController007IPAddress, 57878)

    'Open a Socket to the data stream
    Dim uController007Socket As New Socket(uController007IPAddress.AddressFamily, SocketType.Stream, ProtocolType.Tcp)


    ' ***** uController008 information:

    'Load the IP address of the second uController into uControllerIPAddress
    Dim uController008IPAddress As IPAddress = IPAddress.Parse("192.168.0.8")

    'Combine the IP Adress of the uController and the Port Number into one IPEndPoint variable
    Dim uController008IPandPort As New IPEndPoint(uController008IPAddress, 57878)

    'Open a Socket to the data stream
    Dim uController008Socket As New Socket(uController008IPAddress.AddressFamily, SocketType.Stream, ProtocolType.Tcp)


    ' ***** Input and output storage

    'Input Data Buffer
    Dim InMsg(32) As Byte

    'Output Data Buffer
    Dim OutMsg(32) As Byte

    ' Used to record how many bytes were transmitted
    Dim bytesSent As Integer

    ' Used to record how many bytes were received from ucontrollers
```

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

    Try

      'Establish a TCP/IP Socket connection to uController007 (IP and Port)
      uController007Socket.Connect(uController007IPandPort)

      'Establish a TCP/IP Socket connection to uController008 (IP and Port)
      uController008Socket.Connect(uController008IPandPort)

    Catch ex As Exception

      MessageBox.Show("Socket Connection Error: " & ex.ToString, "Socket Error")

    End Try


    'When reading from the port, should you wait for a response? Or should you not wait?
    'This is set so you do not wait - if nothing is there, move on.
    'Without this line, the timer will lock up.
    uController007Socket.Blocking = False
    uController008Socket.Blocking = False


    'Check if something came in with a timer (every 100 mSeconds)
    Timer1.Interval = 100
    Timer1.Enabled = True


  End Sub


  Private Sub Form1_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing
    'Release the resources when you are done.
    uController008Socket.Close()
    uController007Socket.Close()
  End Sub
```

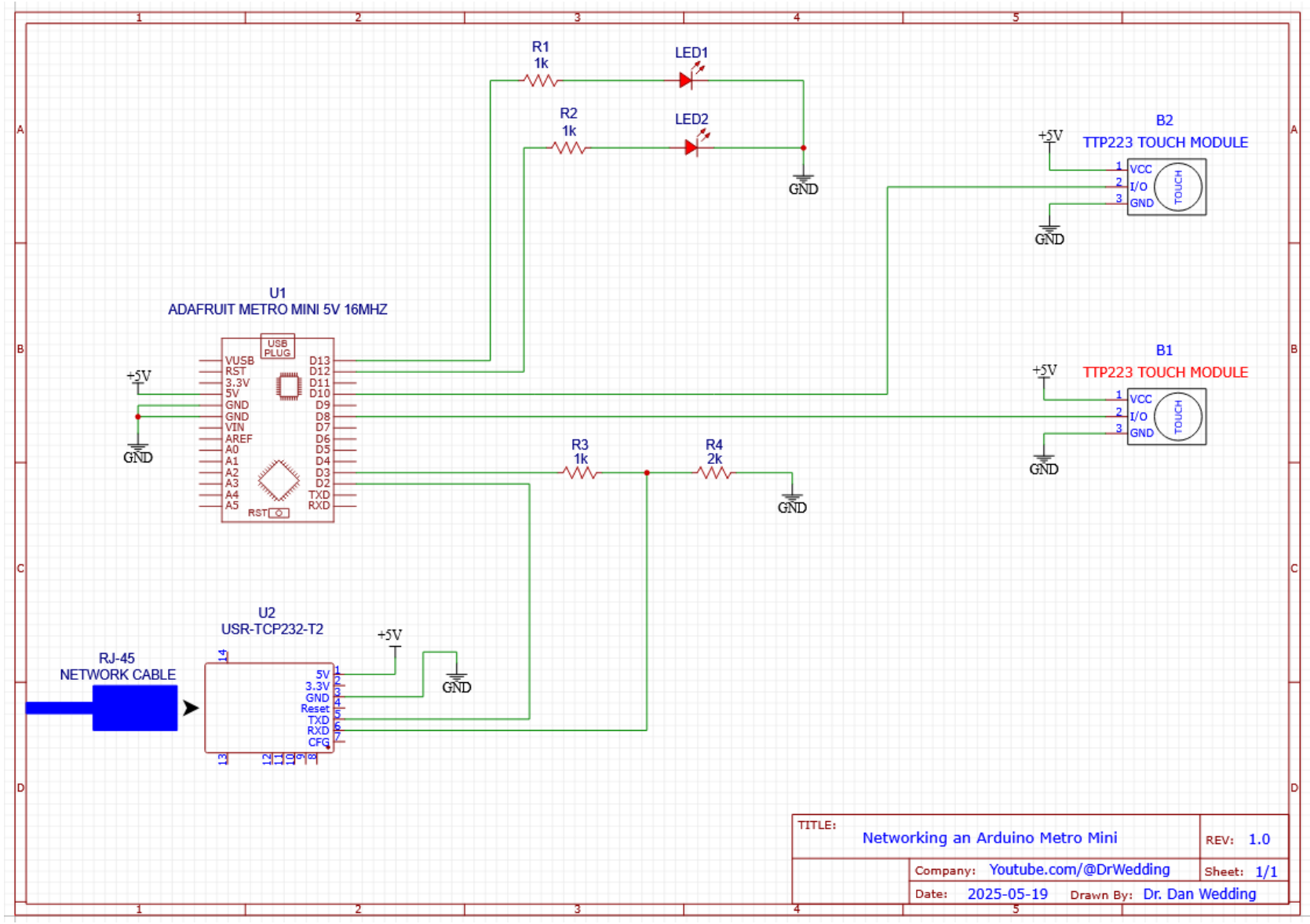
```
Private Sub btn_7_Red_Click(sender As Object, e As EventArgs) Handles btn_7_Red.Click
    Try
        Toggle_Color(1)

        'put the string you want to send into a byte array
        OutMsg = Encoding.ASCII.GetBytes("R")

        'bytesSent is an int that tells you how many bits got sent (If you need it)
        bytesSent = uController007Socket.Send(OutMsg)

    Catch ex As Exception
        MessageBox.Show("Failed to Send to 192.168.0.7", "Transmit Error")
    End Try
End Sub

Private Sub btn_7_Green_Click(sender As Object, e As EventArgs) Handles btn_7_Green.Click
    Try
        Toggle_Color(2)

        'put the string you want to send into a byte array
        OutMsg = Encoding.ASCII.GetBytes("G")

        'bytesSent is an int that tells you how many bits got sent (If you need it)
        bytesSent = uController007Socket.Send(OutMsg)

    Catch ex As Exception
        MessageBox.Show("Failed to Send to 192.168.0.7", "Transmit Error")
    End Try
End Sub
```

```
Private Sub btn_8_Red_Click(sender As Object, e As EventArgs) Handles btn_8_Red.Click
    Try
      Toggle_Color(3)

      'put the string you want to send into a byte array
      OutMsg = Encoding.ASCII.GetBytes("R")

      'bytesSent is an int that tells you how many bits got sent (If you need it)
      bytesSent = uController008Socket.Send(OutMsg)

    Catch ex As Exception
      MessageBox.Show("Failed to Send to 192.168.0.8", "Transmit Error")
    End Try
  End Sub


Private Sub btn_8_Green_Click(sender As Object, e As EventArgs) Handles btn_8_Green.Click
    Try
      Toggle_Color(4)

      'put the string you want to send into a byte array
      OutMsg = Encoding.ASCII.GetBytes("G")

      'bytesSent is an int that tells you how many bits got sent (If you need it)
      bytesSent = uController008Socket.Send(OutMsg)

    Catch ex As Exception
      MessageBox.Show("Failed to Send to 192.168.0.8", "Transmit Error")
    End Try
  End Sub
```

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick

    Try

      If (uController007Socket.Available > 0) Then
        ' Did anything come in from controller 7 (on any port number)?
        ' record how many btyes were received inside bytesreceived / the incoming data was stored in InMsg
        bytesReceived = uController007Socket.Receive(InMsg)

        Toggle_Color(5)

        'convert they bytes in InMsg to chars, one by one. Start at 0 and go to bytesReceived
        rtxt_From_7.AppendText(Encoding.ASCII.GetString(InMsg, 0, bytesReceived))

      End If

    Catch ex As Exception

      MessageBox.Show("Receive Error: " & ex.Message, "ERROR from 192.168.0.7")

    End Try


    '*********************************************************

    Try

      If (uController008Socket.Available > 0) Then
        ' Did anything come in from controller 8 (on any port number) ?
        'record how many btyes were received inside bytesreceived / the incoming data was stored in InMsg
        bytesReceived = uController008Socket.Receive(InMsg)

        Toggle_Color(6)

        'convert they bytes in InMsg to chars, one by one. Start at 0 and go to bytesReceived
        rtxt_From_8.AppendText(Encoding.ASCII.GetString(InMsg, 0, bytesReceived))
      End If


    Catch ex As Exception

      MessageBox.Show("Receive Error: " & ex.Message, "ERROR from 192.168.0.8")

    End Try

  End Sub
```

```
  Sub Toggle_Color(ByRef btn)

    If btn = 1 Then
      If btn_7_Red.BackColor = Color.FromArgb(255, 255, 255, 225) Then
        btn_7_Red.BackColor = Color.FromArgb(255, 255, 192, 192)
      Else
        btn_7_Red.BackColor = Color.FromArgb(255, 255, 255, 225)
      End If

    ElseIf btn = 2 Then
      If btn_7_Green.BackColor = Color.FromArgb(255, 255, 255, 225) Then
        btn_7_Green.BackColor = Color.FromArgb(255, 192, 255, 192)
      Else
        btn_7_Green.BackColor = Color.FromArgb(255, 255, 255, 225)
      End If

    ElseIf btn = 3 Then
      If btn_8_Red.BackColor = Color.FromArgb(255, 255, 235, 190) Then
        btn_8_Red.BackColor = Color.FromArgb(255, 255, 192, 192)
      Else
        btn_8_Red.BackColor = Color.FromArgb(255, 255, 235, 190)
      End If

    ElseIf btn = 4 Then
      If btn_8_Green.BackColor = Color.FromArgb(255, 255, 235, 190) Then
        btn_8_Green.BackColor = Color.FromArgb(255, 192, 255, 192)
      Else
        btn_8_Green.BackColor = Color.FromArgb(255, 255, 235, 190)
      End If

    ElseIf btn = 5 Then
      If Encoding.ASCII.GetString(InMsg, 0, bytesReceived) = "B1 " Then
        rtxt_From_7.SelectionColor = Color.Red
      Else
        rtxt_From_7.SelectionColor = Color.Blue
      End If

    ElseIf btn = 6 Then
      If Encoding.ASCII.GetString(InMsg, 0, bytesReceived) = "B1 " Then
        rtxt_From_8.SelectionColor = Color.Red
      Else
        rtxt_From_8.SelectionColor = Color.Blue
      End If

    End If

  End Sub

End Class
```

ARDUINO SCHEMATIC - Youtube.com/@DrWedding VB25



ARDUINO BOARD PARTS LIST - Youtube.com/@DrWedding VB25

REFERENCE	QUANTITY	DESCRIPTION	PART	NOTE
U1	1	Arduino Uno Compatible	Adafruit.com metro mini 5V / 16Mhz (2590)	**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: An Adafruit Metro Mini (https://www.adafruit.com/product/2590) was used in this video. However, any Arduino Uno compatible microcontroller should work.

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*).

ARDUINO CODE - Youtube.com/@DrWedding VB25

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

```
// Software serial comms (so you can rxd and txd on other pins)
#include <SoftwareSerial.h>

// Software Rxd = pin 2
// software TxD = pin 3

#define rxPin 2
#define txPin 3

// set up a new Software serial port
SoftwareSerial mySerial =  SoftwareSerial(rxPin, txPin);

byte inChar;

int buttonState;
int once1 = 1; // button 1 lockout
int once2 = 1; // button 2 lockout



void setup()

        {
         // put your setup code here, to run once:
        pinMode (8,INPUT);    // Button 1 input
        pinMode (10,INPUT);  // Button 2 input

        // RED LED
        pinMode(12,OUTPUT);
        digitalWrite(12,LOW);

        // GREEN LED
        pinMode(13,OUTPUT);
        digitalWrite(13,LOW);

         // define pin modes for Software Serial tx, rx:
        pinMode(rxPin, INPUT);
        pinMode(txPin, OUTPUT);

         // set the data rate for the SoftwareSerial port
        mySerial.begin(9600);


        }
```

```
void loop()
	{

	delay(10);

	buttonState = digitalRead(8);

	if (buttonState == 1  && once1 == 1)
		{
		mySerial.write("B1 ");
		once1 = 0;
		}
	else if (buttonState == 0  && once1 == 0)
		{
		once1 = 1;
		}

	buttonState = digitalRead(10);

	if (buttonState == 1  && once2 == 1)
		{
		mySerial.write("B2 ");
		once2 = 0;
		}
	else if (buttonState == 0  && once2 == 0)
		{
		once2 = 1;
		}


	while (mySerial.available()>0)
		{
		inChar = mySerial.read();

		if (inChar == 'R')
			{
			// Toggle Red LED
			 digitalWrite(12, !digitalRead(12));
			}

		 else if (inChar == 'G')
			{
			// Toggle Green LED
			digitalWrite(13, !digitalRead(13));
			}

		inChar = ' ';

		}
	}
```