

Digital IO connection for Stepper Controller

Output Signal

Output Name	OutputID	Descriptions of Output
Out1	XINPOS	When reach position/after homing this Output will be ON
Out2	XDIRECTION	Connect to DIR- of the driver, Resistor of 2K is required if source from 24 volts
Out3	XENABLE	Connect to ENA- of the driver, Resistor of 2K is required if source from 24 volts
Out4	YINPOS	When reach position/after homing this Output will be ON
Out5	YDIRECTION	Connect to DIR- of the driver, Resistor of 2K is required if source from 24 volts
Out6	YENABLE	Connect to ENA- of the driver, Resistor of 2K is required if source from 24 volts
Out7	NC	No connection/General Output

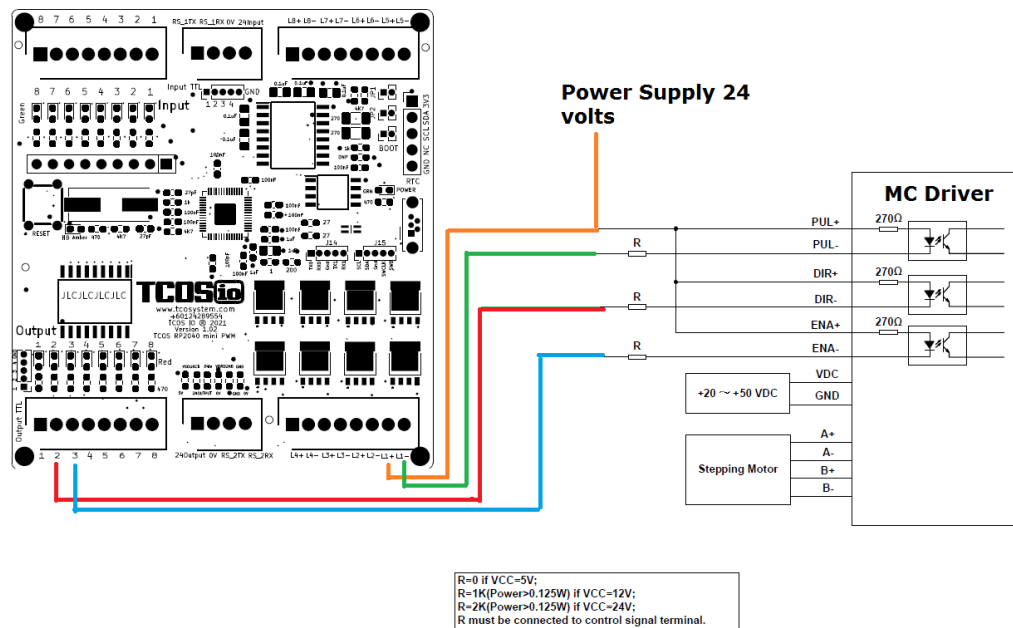
Input Signal

Input Name	InputID	Descriptions of Input
In1	XLMTCW	This sensor is the first sensor to trigger when motor is moving at CW direction
In2	XLMTCCW	This sensor is the first sensor to trigger when motor is moving at CCW direction
In3	XHOME	Optional X Axis Home Sensor, if not present only XLMTCW or XLMTCCW used as Home sensor
In4	XERROR	For Driver that for Error output/Not in used
In5	YLMTCW	This sensor is the first sensor to trigger when motor is moving at CW direction
In6	YLMTCCW	This sensor is the first sensor to trigger when motor is moving at CCW direction
In7	YHOME	Optional Y Axis Home Sensor, if not present only XLMTCW or XLMTCCW used as Home sensor
In8	XERROR	For Driver that for Error output/Not in used

Pulse Signal Connection

PWM Name	PWMID	Descriptions of PWM
L1-	XPUL-	L1- connect to X Axis PUL- Pin for PWM Pulsing, Add resistor for difference driver
L1+	24V	L1+ connect to Power Supply 24 volts,
L2-	YPUL-	L1- connect to X Axis PUL- Pin for PWM Pulsing, Add resistor for difference driver
L2+	24V	L2+ connect to Power Supply 24 volts

Connection Diagram to MC-2705 Microstep Driver for 1 Axis



PWM connection for 6 Channels LED Lighting

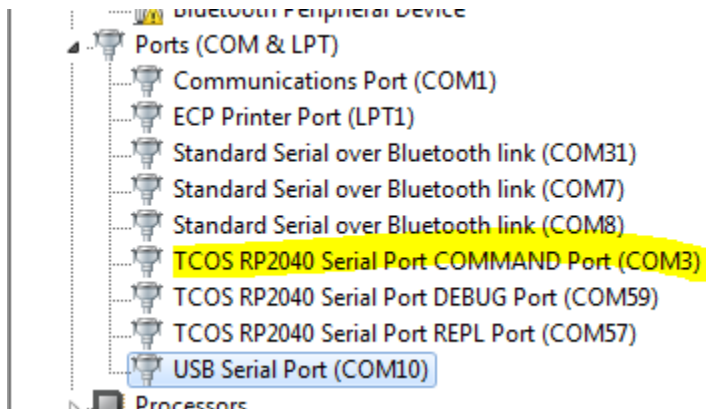
PWM Name	PWMID	Descriptions of PWM
L3-	LED1-	L3- connect to LED1- Pin for PWM Pulsing
L3+	24V and LED2+	L3+ connect to Power Supply 24 volts and LED1+ Pin
L4-	LED2-	L4- connect to LED2- Pin for PWM Pulsing
L4+	24V and LED2+	L4+ connect to Power Supply 24 volts and LED2+ Pin
L5-	LED3-	L5- connect to LED3- Pin for PWM Pulsing
L5+	24V and LED3+	L6+ connect to Power Supply 24 volts and LED3+ Pin
L6-	LED4-	L6- connect to LED4- Pin for PWM Pulsing
L6+	24V and LED4+	L6+ connect to Power Supply 24 volts and LED4+ Pin
L7-	LED5-	L7- connect to LED5- Pin for PWM Pulsing
L7+	24V and LED5+	L7+ connect to Power Supply 24 volts and LED5+ Pin
L8-	LED6-	L8- connect to LED6- Pin for PWM Pulsing
L8+	24V and LED6+	L8+ connect to Power Supply 24 volts and LED6+ Pin

Driver Installation

Driver can be download from this link below

<https://github.com/cgtan2020/RP2040-miniPLC-PWM-0808/tree/main/driver>

Download and copy the 3 Drivers files to your computer, Plug in the RP2040 miniPLC – 2 Axis Stepper Controller and select the respective driver for to emulate CDC port on your PC.



There will be 3 new TCOS RP2040 Serial Port, the one that can be use to control the board is with the name COMMAND PORT. The default baud rate is 115200,8,N,1

Control command protocol

Summary table of the Protocol to control the RP2040 miniPLC – 2 Axis Stepper Controller

Each command must start with @ and end with *, start with command code

Command Code	Example	Parameter	Function Details
Homing			
XHM	@XHM,15000,CW,0,60000,*	1 st Parameter is the homing speed, in hertz. 2 nd Parameter is the direction of homing, CW or CCW. 3 rd Parameter is Not in used 4 th Parameter is Not in used	X Homing Function, successful function execution it will reply with @X,HOME DONE,*\n. XINPOS output will be ON after homing completed
YHM	@YHM,15000,CW,0,60000,*	1 st Parameter is the homing speed, in hertz. 2 nd Parameter is the direction of homing, CW or CCW. 3 rd Parameter is Not in used 4 th Parameter is Not in used	Y Homing Function, successful function execution it will reply with @Y,HOME DONE,*\n. YINPOS output will be ON after homing completed
Jogging			
XJG	@XJG,2000,CW,17,*	1 st Parameter is the jog speed, in hertz. 2 nd Parameter is the direction of homing, CW or CCW. 3 rd Parameter is the step to jog	X Jogging Function, successful function execution it will reply with @X,JOG DONE,*\n.
YJG	@YJG,2000,CW,10,*	1 st Parameter is the jog speed, in hertz. 2 nd Parameter is the direction of homing, CW or CCW. 3 rd Parameter is the step to jog	Y Jogging Function, successful function execution it will reply with @Y,JOG DONE,*\n.

Move Relative			
XMR	@XMR, 25000, CW, 1000, *	1 st Parameter is the jog speed, in hertz. 2 nd Parameter is the direction of homing, CW or CCW. 3 rd Parameter is the step to move	X Move Relative Function, successful function execution it will reply with @X, MOVE DONE, * \n. XINPOS output will be ON after homing completed
YMR	@YMR, 25000, CCW, 1000, *	1 st Parameter is the jog speed, in hertz. 2 nd Parameter is the direction of homing, CW or CCW. 3 rd Parameter is the step to move	Y Move Relative Function, successful function execution it will reply with @Y, MOVE DONE, * \n. YINPOS output will be ON after homing completed
Move Absolute			
XMA	@XMA, 30000, 5000, *	1 st Parameter is the jog speed, in hertz. 2 nd Parameter is the absolute position to move to	X Move Relative Function, successful function execution it will reply with @X, MOVE DONE, * \n. XINPOS output will be ON after homing completed
YMA	@YMA, 30000, -5000, *	1 st Parameter is the jog speed, in hertz. 2 nd Parameter is the absolute position to move to	Y Move Relative Function, successful function execution it will reply with @Y, MOVE DONE, * \n. YINPOS output will be ON after homing completed
Other Function			
	Query Heart Beat		
QHB	@QHB*	Get the RP2040 time in seconds	Reply will be @QHB, 323.12342, *
	Query Current Position		
QCP	@QCP*	Get the motor current position, when all motion is stop	Reply will be @QCP, 3231, 1342, *

	Acceleration Deceleration Profile		
ADP	@ADP, 5, 10, 10, 12, *	1 st Parameter is the X Acceleration Percent. 2 nd Parameter is the Y Acceleration Percent. 3 rd Parameter in the X Deceleration Percent. 4 th Parameter is the Y Deceleration Percent.	Reply @ADP, DONE, *\n"
	Soft Reboot RP2040		
SRT	@SRT*	No Parameter	Reply @SRT, OK, *\n"
	6 channel of PWM for LED Lighting		
SI	Set Intensity	Command to send is @SIXFFFF*, X is the channel value from 0-5, FFFF is the 32 bit hexadecimal value from 0-65535, 16 bit integer	Reply system will reply @SIX, OK*, where X is the channel value from 0 to 5

Sample program to control the RP2040 miniPLC – 2 Axis Stepper Controller in VB6

<https://github.com/cgtan2020/RP2040-miniPLC-PWM-0808/tree/main/SampleCode>

```
'Global variable and functions
Option Explicit

Global Const STX = 64
Global Const ETX = 42
Global Const CR = &HD

Global str1 As String
Global str2 As String
Global gCount As Long
Global gSentStr(10, 10) As String
Global gRecStr As String

Global gstrSerial As String
Global gstrAddSerial As String

Global gbGotSTX As Boolean
Global gbGotETX As Boolean
Global giSTXCount As Integer
Global giETXCount As Integer
Global giXIndex As Integer
Global giYIndex As Integer
Global gbXDone As Boolean
Global gbYDone As Boolean
Global gbTrigger As Boolean

Public Declare Sub Sleep Lib "Kernel32" (ByVal dwMilliseconds As Long)

Public Function ProcessSerialPort() As Boolean
'return true if can process the port else keep appending
'Global gbGotSTX As Boolean
'Global gbGotETX As Boolean
'Global giSTXCount As Integer
'Global giETXCount As Integer

Dim leftover As String
Dim temp As String
Dim pos As Integer
Dim pos2 As Integer
Dim length As Integer
Dim length2 As Integer

leftover = gstrAddSerial

giSTXCount = 0
giETXCount = 0

pos = InStr(leftover, Chr(STX))

While pos
    giSTXCount = giSTXCount + 1
    length = Len(leftover)

    If (pos < length) Then

        temp = Right(leftover, length - pos - 1)
        leftover = temp
        pos = InStr(leftover, Chr(STX))
    Else
        pos = 0
    End If

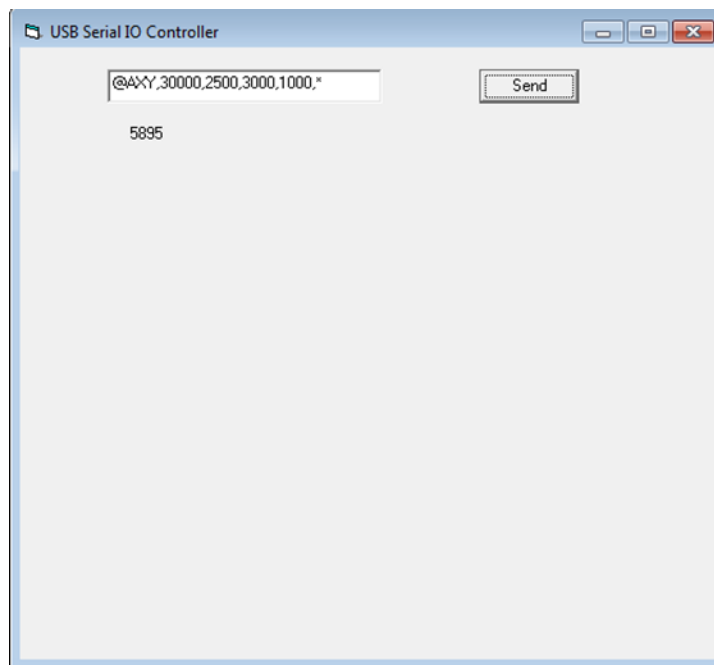
Wend
```

```
leftover = gstrAddSerial
pos2 = InStr(leftover, Chr(ETX))

While pos2
    giETXCount = giETXCount + 1
    length = Len(leftover)
    If (pos2 < length) Then
        temp = Right(leftover, length - pos2 - 1)
        leftover = temp
        pos2 = InStr(leftover, Chr(ETX))
    Else
        pos2 = 0
    End If
Wend

If (giETXCount = giSTXCount) Then
    ProcessSerialPort = True
Else
    ProcessSerialPort = False
End If

End Function
```




```

'Main Program
Option Explicit

Private Sub cmdSend_Click()
MSComm1.Output = txtSend.Text

End Sub

Private Sub Form_Load()
Dim x As Integer
Dim y As Integer

MSComm1.PortOpen = True

For x = 0 To 9
    For y = 0 To 4
        gSentStr(x, y) = "@AXY,30000,2500," + CStr(x * 2000) + "," + CStr(y * 1000) + ",*"
    Next y
Next x

giXIndex = 0
giYIndex = 0
gbXDone = False
gbYDone = False

gCount = 0

End Sub

Private Sub Form_Unload(Cancel As Integer)
MSComm1.PortOpen = False

End Sub

Private Sub MSComm1_OnComm()
Dim mydata As String
Dim direction As Integer

Select Case MSComm1.CommEvent
Dim bret As Boolean

    ' Handle each event or error by placing
    ' code below each case statement

    ' Errors
    Case comEventBreak      ' A Break was received.
    Case comEventFrame      ' Framing Error
    Case comEventOverrun    ' Data Lost.
    Case comEventRxOver     ' Receive buffer overflow.
    Case comEventRxParity   ' Parity Error.
    Case comEventTxFull     ' Transmit buffer full.
    Case comEventDCB        ' Unexpected error retrieving DCB]

    ' Events
    Case comEvCD            ' Change in the CD line.
    Case comEvCTS           ' Change in the CTS line.
    Case comEvDSR           ' Change in the DSR line.
    Case comEvRing          ' Change in the Ring Indicator.
    Case comEvReceive       ' Received RThreshold # of
                            ' chars.

    gstrSerial = MSComm1.Input

    gstrAddSerial = gstrAddSerial + gstrSerial

    'Process the return string for the read data

    bret = ProcessSerialPort()

```

```

If bret Then

    Debug.Print "Got reply ->"; gstrAddSerial

    If InStr(gstrAddSerial, "@X,MOVE DONE,*") Then
        Debug.Print gstrAddSerial
        gbXDone = True
        gstrAddSerial = ""
        gstrSerial = ""
    End If

    If InStr(gstrAddSerial, "@Y,MOVE DONE,*") Then
        Debug.Print gstrAddSerial
        gbYDone = True

        gstrAddSerial = ""
        gstrSerial = ""
    End If

    If (gbXDone = True And gbYDone = True) Then

        gCount = gCount + 1
        Labell1.Caption = CStr(gCount)
        gbTrigger = True

        If giYIndex > 3 Or giYIndex < 0 Then
            giYIndex = 0
        End If

        If (giYIndex Mod 2 = 0) Then

            direction = 1
        Else
            direction = -1
        End If

        giXIndex = giXIndex + direction

        If giXIndex > 9 Or giXIndex < 0 Then
            giYIndex = giYIndex + 1
        End If

        If giXIndex > 9 Or giXIndex < 0 Then
            giXIndex = 0
        End If

        gbXDone = False
        gbYDone = False
        'Sleep (250) 'processing

    End If

End If

Case comEvSend ' There are SThreshold number of
                ' characters in the transmit
                ' buffer.
Case comEvEOF ' An EOF charater was found in
               ' the input stream

End Select
End Sub

Private Sub Timer1_Timer()

    If gbTrigger = True Then
        MSComm1.Output = (gSentStr(giXIndex, giYIndex))
        gbTrigger = False
    End If

End Sub

```