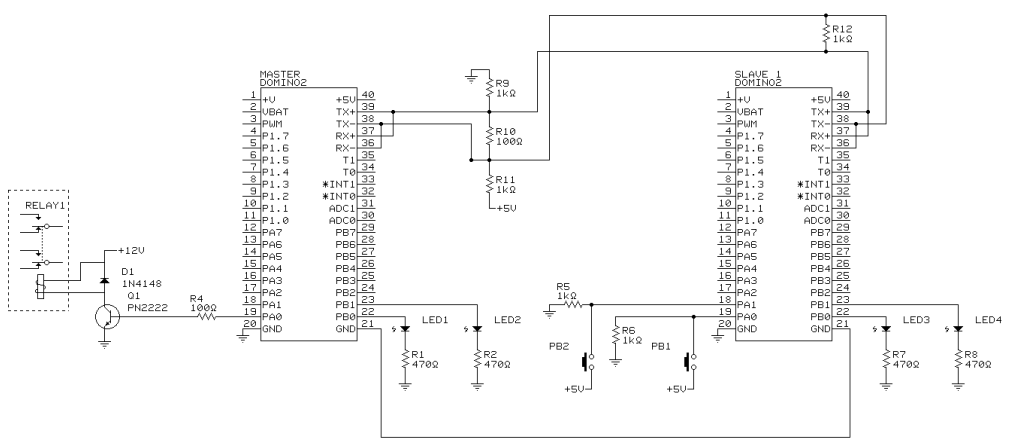
	<h1>Application Note</h1>
	<h2>Product: Domino 1 or 2</h2>
<h3>Networking 2 Domino's Via RS-485</h3>	<h3>Date: 8/26/99</h3>
<p>Introduction: This application note demonstrates how to network 2 Domino controllers together in a master/slave protocol.</p>	
<p>Background: Multiple Domino's can be used in a network using a single twisted pair of wire for communication. Network protocol requires that only one unit be allowed to transmit at any given time. All other units are in receiving mode. This is known as master/slave protocol.</p> <p>This is accomplished by requiring a Domino or a device like a PC to be the net master. The master talks to any slave unit by passing information to it or requesting information from it, The slaves must never answer the master until a response is requested. The master then relinquishes the net to that slave for the response and regains it when the slave is finished. This arrangement enables multiple controllers to work together gathering numerous inputs and controlling innumerable outputs, limited only by the system size.</p>	
<p>How it works: This application note shows 2 Domino 2's connected via RS-485 using the master/slave protocol. The following schematic demonstrates how to connect the two together.</p>  <p>The master transmits the slave's name each time it asks the slave a question. In the program listing the slaves name is the number 1. Each time the master transmits, it turns off its transmitter and goes into receive mode. In the beginning of both the master and the slave programs we poke two assembly routines into memory using the XBY command. We then call these routines to enable the transmitter or disable it. The slave is in receive mode from the start of its program. The only time it goes into transmit mode is after it receives a command from the master.</p>	

In this application note the master asks the slave to poll its inputs for a high and uses the PN2222 as a switch to power the relays coil. When PB2 is pressed the slave tells the master to switch the transistor on and when PB1 is pressed it tells the master to switch the transistor off. Although this application note only connects 2 Domino 2's together it is very easy to add a third Domino 2 by changing the master program to execute different routines for the new slave. The slave program is identical except for the slaves name.

Program Listing:

```
10 REM *** Master
20 REM *** The master receives input from Slave 1
30 REM *** Relay 1(PA0) is controlled by Slave 1.
40 REM *** Relay 2(PA1) is controlled by Slave 2.
50 REM *** LED1(PB0) is lit when there is power to the Master.
60 REM *** LED2(PB1) is lit whenever the transmitter is enabled.
70 REM *** Subroutine Menu
80 REM *** ADDRESS FUNCTION
90 REM *** 11000 Coprocessor Write Routine
100 REM *** Transmit Disable Assembly Instructions
110 XBY(32000)=0C2H
120 XBY(32001)=0B4H
130 XBY(32002)=022H
140 REM *** Transmit Enable Assembly Instructions
150 XBY(32003)=0D2H
160 XBY(32004)=0B4H
170 XBY(32005)=022H
180 REM *** Setting up the Coprocessor.
190 REM *** Setting Port A as outputs.
200 R=30H : DT=0H
210 GOSUB 11000
220 REM *** Setting Port B as outputs.
230 R=33H : DT=0H
240 GOSUB 11000
250 REM *** Setting PB0 high. Indicates the Master has power.
260 R=35H : DT=1H
270 GOSUB 11000
280 REM *** Enable the transmitter
290 CALL 32003
300 REM *** Set PB1 high. Indicates that the transmitter is on.
310 R=35H : DT=3H
320 GOSUB 11000
330 REM Start of a delay to wait for the slave to get set up.
340 FOR Z=0 TO 100 : NEXT Z
350 ?"1",CR
360 REM *** Disable the transmitter.
370 CALL 32000
380 REM *** Set PB1 low. This indicates the transmitter is disabled.
390 R=35H : DT=1H
400 GOSUB 11000
410 INPUT A,B
420 REM *** Push button 2 has been pressed.
430 IF (A=1.AND.B=2) THEN GOTO 500
440 REM *** Push button 1 has been pressed
450 IF (A=1.AND.B=1) THEN GOTO 550
460 REM *** No push buttons were pushed or both were pushed.
470 IF (A=1.AND.B=0) THEN GOTO 610
480 REM *** Coprocessor routines for when a push button is pressed.
490 REM *** Set PA0 high. Relay 1 turns on.
```

```

500 R=50H : DT=1H
510 GOSUB 11000
520 REM *** Poll Slave 1 again.
530 GOTO 290
540 REM *** Set PA0 low. Relay 1 turns off
550 R=50H : DT=0H
560 GOSUB 11000
570 REM *** Poll Slave 1 again.
580 GOTO 290
590 REM *** PA0 stays in the same state. No push buttons were
600 REM *** pressed or both were pressed.
610 GOTO 290
10997 REM *****
10998 REM Coprocessor Write Routine
10999 REM
11000 PUSH 2000H+R,DT
11010 CALL 0F128H
11020 POP DT
11030 IF DT>255 THEN PRINT "Communications error !!!"
11040 RETURN
11050 REM
11060 REM *****

10 REM *** Slave 1
20 REM *** The slave reads it's inputs on Port A and sends
30 REM *** it's data to the Master. When push button 1 (PA0 pin 19)
40 REM *** is pressed, relay 1 of the Master disengages.
50 REM *** When push button 2 (PA1 pin 18)is pressed, relay 1
60 REM *** of the Master engages.
70 REM *** LED3 is on when Slave 1 has power.
80 REM *** LED4 is on whenever Slave 1's transmitter is disabled.
90 REM *** Subroutine Menu
100 REM *** ADDRESS FUNCTION
110 REM *** 10030 Coprocessor Read Routine
120 REM *** 11000 Coprocessor Write Routine
130 REM *** Transmit Disable Assembly Instructions
140 XBY(32000)=0C2H
150 XBY(32001)=0B4H
160 XBY(32002)=022H
170 REM *** Transmit Enable Assembly Instructions
180 XBY(32003)=0D2H
190 XBY(32004)=0B4H
200 XBY(32005)=022H
210 REM *** Setting up the Coprocessor
220 REM *** Setting Port A as inputs.
230 R=30H : DT=1H
240 GOSUB 11000
250 REM *** Setting Port B as outputs.
260 R=33H : DT=0H
270 GOSUB 11000
280 REM *** Setting PB0 high. Indicates the Slave has power.
290 R=35H : DT=1H
300 GOSUB 11000
310 REM *** Disabling Transmit
320 CALL 32000
330 REM *** Setting PB1 low. Indicates transmit disabled.
340 R=35H : DT=1H
350 GOSUB 11000
360 REM *** Read Port A's inputs and store them in a variable

```

```

370 N=0 : M=0
380 REM *** Reading PA0 (pin 19) and storing the value in N.
390 R=40H
400 GOSUB 10030
410 N=DT
420 REM *** Reading PA1 (pin 20) and storing the value in M
430 R=41H
440 GOSUB 10030
450 M=DT
460 REM *** Waiting for input from Master
470 INPUT A
480 IF A<>1 THEN GOTO 470
490 REM *** Enabling Transmit
500 CALL 32003
510 REM *** Setting PB1 (pin 23) high. Indicates transmit enabled
520 R=35H: DT=3H
530 GOSUB 11000
540 REM *** Push button 2 has been pressed and 1 hasn't.
550 IF (M=1.AND.N=0) THEN PRINT "1,2", CR ELSE GOTO 590
560 REM *** Read the inputs again.
570 GOTO 320
580 REM *** Push button 1 has been pressed and 2 hasn't.
590 IF (M=0.AND.N=1) THEN PRINT "1,1", CR ELSE GOTO 630
600 REM *** Read the inputs again.
610 GOTO 320
620 REM *** No push buttons were pushed.
630 IF (M=0.AND.N=0) THEN PRINT "1,0", CR ELSE GOTO 690
640 REM *** Read the inputs again.
650 GOTO 320
660 REM *** Push button 1 and 2 were pressed.
670 REM *** If both push buttons are pressed then it acts like
680 REM *** no push buttons were pressed.
690 IF (M=1.AND.N=1) THEN PRINT "1,0", CR
700 REM *** Read the inputs again}
710 GOTO 320
10000 REM *****
10010 REM Coprocessor Read Routine
10020 REM
10030 PUSH 2000H + R
10040 CALL 0F12Ch
10050 POP DT
10060 IF DT>255 THEN ?"Communications error!!!"
10070 RETURN
10080 REM
10997 REM *****
10998 REM Coprocessor Write Routine
10999 REM
11000 PUSH 2000H+R,DT
11010 CALL 0F128H
11020 POP DT
11030 IF DT>255 THEN PRINT "Communications error !!!"
11040 RETURN
11050 REM
11060 REM *****

```