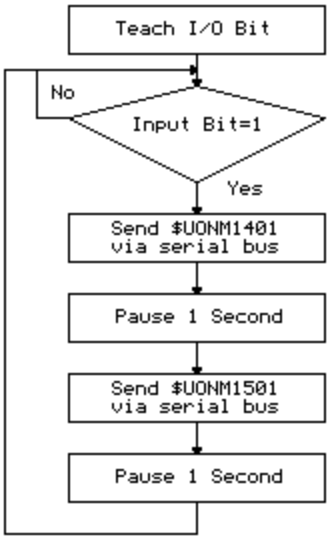
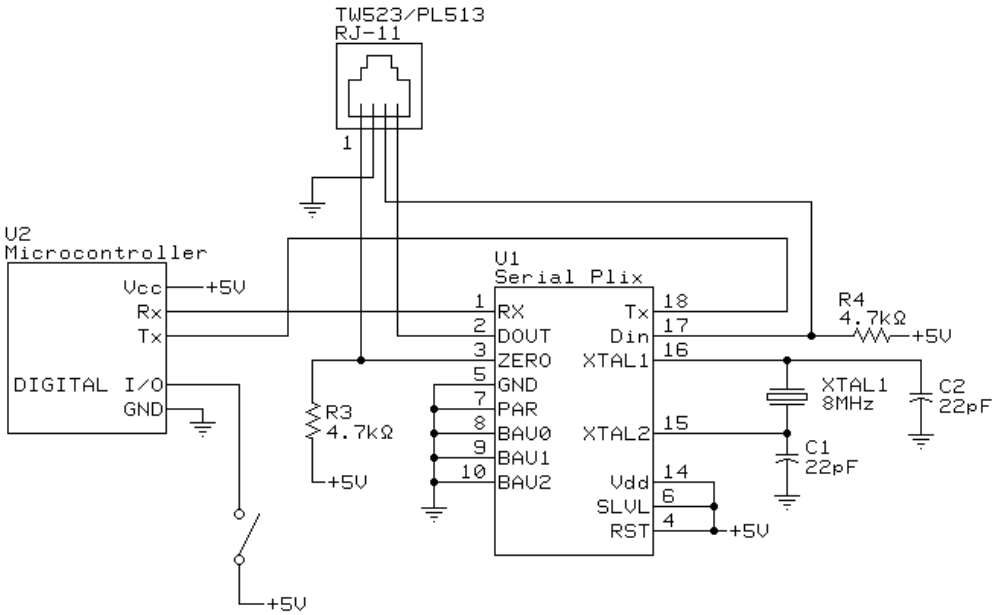
Micromint, Inc	**Application Note**
	Product: Domino, Serial Plix
Transmitting and Receiving Data over AC Power Lines.	Date: 6/15/99

Introduction: Transmitting data over power lines using Serial Plix can be very simple. The following application note demonstrates how to accomplish this by using any two micro-controllers with the two Serial Plix chips.

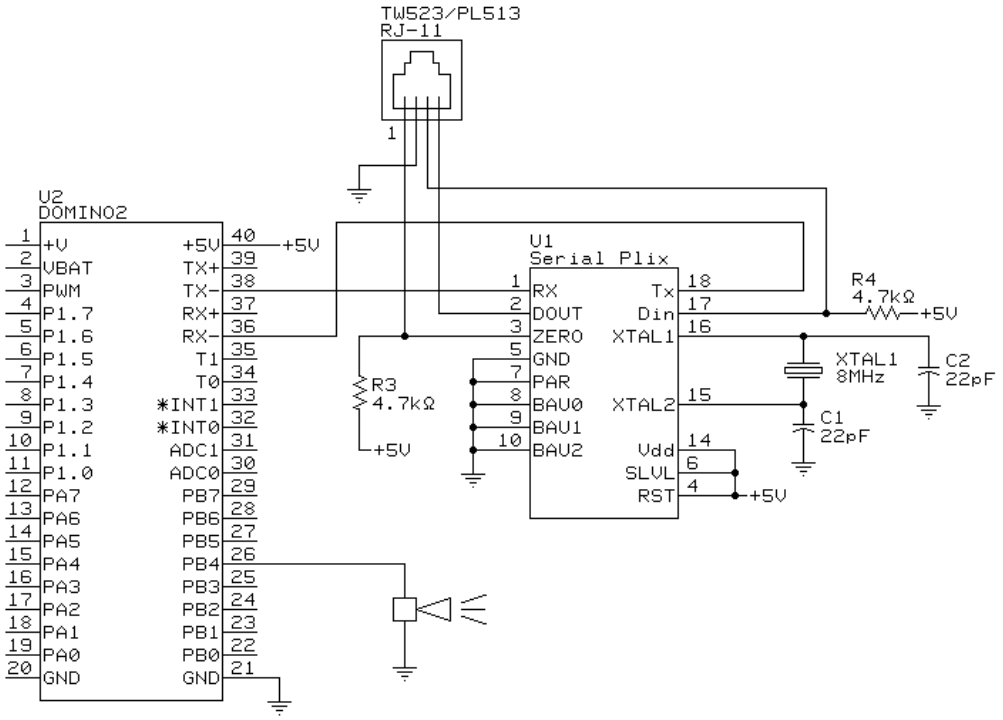
Background: The Serial Plix is an 18-pin CMOS chip, which provides an intelligent communication interface between a serial port and X-10 AC power-line control modules. Using the Serial Plix and a TW523 to transmit and receive data, it is very easy to have a micro-controller perform a specific task remotely.

How it works: This application note demonstrates a generic micro-controller transmitting a series of commands to the Serial Plix and having it transmit M14 followed by M15, when a door opens. The code sequence of M14 and M15 is user defined. The Domino is used to tell the Serial Plix when to look for data on the power lines and if the Domino receives M14 followed by M15 from the Serial Plix, the Domino then sets an output bit high to sound a piezo buzzer. The following flow chart and schematic show the necessary steps and connections required for any micro-controller to monitor the door.





The following schematic is the connections needed for the Domino 2 to operate as the receiver.



Program Listing:

```
10 REM*** This program demonstrates receiving data
20 REM*** over the power lines.
30 MTOP=12288
```

```
 40 REM *** Allocate memory for one 4-byte string.
 50 STRING 6,4
 60 REM *** Set PB4 as an output.
 70 PUSH 2033H,EFH
 80 CALL 0F128H
 90 POP C
100 REM *** Coprocessor error checking
110 IF C<=255 THEN  GOTO 160
120 REM *** Setting P1.7 and P1.6 high masking the rest of the port.
130 PORT1=PORT1 .OR. 0C0H
140 GOTO 70
150 REM *** Set PB4 low
160 PUSH 205CH,0
170 CALL 0F128H
180 POP C
190 REM *** Coprocessor error checking.
200 IF C<=255 THEN  GOTO 240
210 PORT1=PORT1 .OR. 0C0H
220 GOTO 160
230 REM *** Transmit to the Serial Plix.
240 ?"$DINA9950",
250 REM *** Receive from the Serial Plix.
260 INPUT $(0)
270 REM *** Taking the House and Unit code from the Serial Plix
280 REM *** and storing them into variables A, B, and C.
290 REM *** To figure out where the variables are in memory
300 REM *** you subtract the total bytes expressed in the
310 REM *** string from MTOP. In this situation it is
320 REM *** 12288-6=12282. Since we don't need to store the
330 REM *** lead in character(!) we start at 12283.
340 A=XBY(12283)
350 B=XBY(12284)
360 C=XBY(12285)
370 IF (A<>77.OR.B<>49.OR.C<>52) THEN  GOTO 240
380 REM *** Transmit to the Serial Plix.
390 ?"$DINA9950",
400 REM *** Receive from the Serial Plix.
410 INPUT $(0)
420 A=XBY(12283)
430 B=XBY(12284)
440 C=XBY(12285)
450 IF (A<>77.OR.B<>49.OR.C<>53) THEN  GOTO 390
460 REM *** Set PB4 high.
470 PUSH 205CH,1
480 CALL 0F128H
490 POP C
500 REM *** Coprocessor error checking
510 IF C<=255 THEN  GOTO 550
520 PORT1=PORT1 .OR. 0C0H
530 GOTO 470
540 REM *** Delay the program for 1 second.
550 TIME=0 : CLOCK1
560 IF TIME<1 THEN  GOTO 560
570 CLOCK0
580 GOTO 160
```