


|   |                             |
|---|-----------------------------|
|    | <b>Application<br/>Note</b> |
|   | <b>PicStic 4</b>            |
| <b>Using the PASS routine to<br/>communicate with the<br/>coprocessor</b>   | <b>11/20/00</b>             |
| <p><b>Introduction:</b> The PicStic 4 is a low-cost, industrial-oriented controller with a built-in intelligent I/O coprocessor. A routine called PASS, was written to allow the processor to communicate with the coprocessor.</p>   |                             |
| <p><b>Background:</b> The PicStic 4 coprocessor has many features that make the PicStic 4 so versatile. These features include 4-channels 8-bit ADC, 2K EEPROM data storage, 8 digital I/O and routines for keypad scanning and LCD controls. These are only a few of the feature that the PicStic 4 has to offer. The call PASS routine was written to make it easy for the user to take advantage of all the features that the coprocessor has available.</p>   |                             |
| <p><b>How it works:</b> There are three variables used to pass and receive all data to and from the I/O coprocessor. The communication between the processors runs at 62.5 Kbps. To send a command and receive a reply takes less than 1mS. The commands that are sent to the coprocessor are always three bytes in length. The first byte (variable B19) is the command byte. The second byte (variable B20) is the LSByte of a 16-bit parameter. The third byte (variable B21) is a MSBtye of the 16-bit parameter. Command bytes, which require no parameter, are padded with null data. Section 7.0 in the PicStic 4 data sheet lists and describes the coprocessor's command set.</p> <p>Once the command and the parameter is set, the user calls the PASS routine to execute the command. After the processor executes the PASS routine a reply is stored in B20 and B21. The command that was executed is echoed back into B19 during the reply. Commands that don't require replies return the value FF into B20 and B21. If the reply stored in B20 and B21 is 00 and the command is not echoed, then an error occurred in the transmission.</p> <p>The program below illustrates how a user could input a command and the necessary parameter. The program also shows a simple form of error checking. The Program is written is PicBasic.</p> |                             |
| <p><b>Program Listing:</b></p> <pre> start:  SEROUT 7,N9600,("Enter a command # and two parameters",10,13)         SEROUT 7,N9600,("      COMMAND #? ")         SERIN 6,N9600,#B19 : SEROUT 7,N9600,(#B19,10,13)         SEROUT 7,N9600,("      PARAMETER LSB? ")         SERIN 6,N9600,#B20 : SEROUT 7,N9600,(#B20,10,13)         SEROUT 7,N9600,("      PARAMETER MSB? ")         SERIN 6,N9600,#B21 : SEROUT 7,N9600,(#B21,10,13)          CALL PASS </pre>  |                             |

```
        if B19=0 and B20=0 and B21=0 then error

data:   SEROUT 7,N9600,("I have received back these three bytes",10,13)
        SEROUT 7,N9600,(#B19," ",#B20," ",#B21,10,13,10,13)

        goto start

error:   SEROUT 7,N9600,("There has been an error communicating",10,13)
        goto data
```