

## Implementing Table Read and Table Write

### INTRODUCTION

This application brief discusses how to read data from program memory to data memory and write data from data memory to program memory.

### RETLW K Instruction

As in all PIC17CXX family parts, the simplest method used to retrieve data from program memory to data memory is to use the RETLW K instruction. For example:

```

; simple program to transfer
; table values to PortB
Main
    movlw 5,W      ;load offset
    call SimpleTableRead
    movwf PortB    ;output to PortB
    .
    .
    .
SimpleTableRead
    addwf PC       ;add offset to PC
Table retlw 0     ;return a known
                  ;table value based
                  ;on the OFFSET.
    .
    .
    .
    retlw 10

```

In the example above, OFFSET is loaded with the required offset to the Table and the subroutine SimpleTableRead is called. The table value is returned in the W register. In this manner program memory can be transferred to data memory.

### Table Read Instruction

The PIC17C42 has an expanded instruction set which includes the TABLRD and TLRD instructions. These instructions are specifically constructed to transfer data from program memory to data memory.

If the instruction syntax is: TABLRD t, i, f, the sequence in which this instruction is executed is as follows:

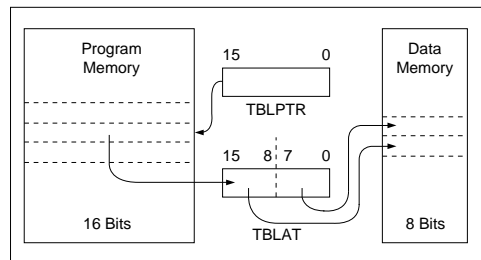
- if t = 1 then the high byte of the table latch (TBLATH) is loaded in the file register f.
- else (if t = 0) the low byte of the table latch (TBLATL) is loaded in the file register f.
- next, the 16 bit data pointed to by the table pointer (TBLPTR) is loaded into the table latch.
- lastly, if i = 1 the table pointer (TBLPTR) is incremented.

**Note:** The first time this instruction is executed in a sequence, the table latch will not be initialized, hence an unknown value will be loaded in the file register. This is not a problem if the user overwrites the same f register in the next subsequent instruction.

If the instruction syntax is: TLRD t, f, the sequence in which this instruction is executed is as follows:

- if t = 1 then the high byte of the table latch (TBLATH) is loaded in the file register f.
- else (if t = 0) the low byte of the table latch (TBLATH) is loaded in the file register f.

**FIGURE 1 - TABLE READ**



# Implementing Table Read and Write in PIC17C42

## Read In Line

A simple method of transferring data from program memory to data memory is to use the `TABLRD` and `TLRD` instruction in sequence as shown in the example below:

```
;transfer 6 bytes of data in program memory at
;0x500, to data memory at 0x80:
```

ReadInLine

```
    movlw 05          ;load table pointer with
    ; 0x500

    movwf TBLPTRH    ;
    clrf TBLPTRL     ;

    tablrd0,1,0x80   ;get 16 bit value in
    ;table latch.

    tlrld 0,0x80     ;low byte (1st) @ 80
    tablrd1,1,0x81   ;high byte (2nd) @ 81
    tlrld 0,0x82     ;3rd byte @ 82
    tablrd1,1,0x83   ;4th byte @ 83
    tlrld 0,0x84     ;5th byte @ 84
    tablrd1,1,0x85   ;6th byte @ 85
```

## Reading a Block of Data

In instances where a block of N bytes needs to be transferred from program memory to data memory, the `TABLRD` and `TLRD` instruction need to be included in a loop which checks for N transfers.

```
;transfer 'COUNT' bytes (even values only) of
;data at program memory 'MESSAGE' to data memory
;at:
```

; 'RAM\_BUFFER'

ReadBlock

```
    movlw high MESSAGE ;load table pointer
    movpf W,TBLPTRH    ;
    movlw low MESSAGE  ;
    movpf W,TBLPTRL    ;

    bcf ALUSTA,5       ;enable post auto
    ;increment of FSR0

    movlw RAM_BUFFER   ;initialize FSR0 to
    ;RAM_BUFFER

    movfp W,FSR0       ;
    movlw COUNT/2      ;initialize count
    tablrd 1,1,RAM_BUFFER ;initialize table
    ;latch
```

ReadBlockLoop

```
    tlrld 1,0x00       ;do indirect read of
    ;high byte

    tablrd 0,1,0x00    ;do indirect read of
    ;low byte

    decfsz W            ;check if count = 0
    goto ReadBlockLoop ;no then do next
    return              ;else end of
    ;transfer.
```

Program	Code Size	Transfer Rate
Simple Table Read (using RETLW)	$N + 3$	6 cycles/byte
Read In-Line	$4 + N + N/2$	1.5 cycles/byte
Read Block (using loop)	$14 + N/2$	3 cycles/byte

N = Number of bytes to transfer

	Code size		
	Simple Table Read	Read In-Line	Read Block
N = 10	13	19	19
N = 20	23	34	24

Conclusion:

In cases where the number of bytes to be transferred is small, the Read In-Line offers small code size for fast transfer rate. However, as the number of bytes to be transferred increases, the Read Block offers optimum code size for a decent transfer rate.

# Implementing Table Read and Write in PIC17C42

## Table Write Instruction

The PIC17C42 has a `TABLWT` and a `TLWT` instruction which transfer data from data memory to program memory. Note in cases where the table pointer points to internal EPROM, the table write instruction will try to program the EPROM, hence the programming voltage must be present on the `VPP` line to successfully program the part.

The instruction syntax is: `TABLWT t, i, f`.

The sequence in which this instruction is executed is as follows:

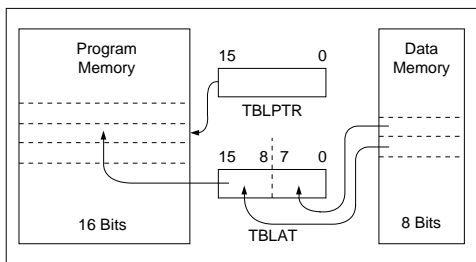
- if `t = 1` then the file register `f` is loaded to the high byte of the table latch (`TBLATH`).
- else (if `t = 0`) the file register `f` is loaded to the low byte of the table latch (`TBLATL`).
- next, the 16 bit data in the table latch is transferred to the program memory pointed to by the table pointer (`TBLPTR`).
- lastly, if `i = 1` the table pointer (`TBLPTR`) is incremented.

The instruction syntax is: `TLWT t, f`

The sequence in which this instruction is executed is as follows:

- if `t = 1` then the file register `f` is loaded to the high byte of the table latch (`TBLATH`).
- else (if `t = 0`) the file register `f` is loaded to the low byte of the table latch (`TBLATL`).

FIGURE 2 - TABLE WRITE



## Write in Line

A simple method of transferring data from data memory to program memory is to use the `TABLWT` and `TLWT` instruction in sequence as shown in the example below:

```
;transfer 6 bytes of data in data memory at 0x80,  
to ;program memory at 0x5000:
```

```
ReadInLine  
    movlw 50           ;load table pointer with  
                        ;0x5000  
  
    movwf TBLPTRH     ; /  
    clrf TBLPTRL     ; /  
    tlwt 1,0x80       ;high byte @ table latch.  
    tablwt 0,1,0x81   ;low byte @ table latch;  
                        ;latch @ prog. mem.  
  
    tlwt 1,0x82       ;3rd and 4th byte @ prog.  
                        ;mem.  
  
    tablwt 0,1,0x83   ; /  
    tlwt 1,0x84       ;5th and 6th byte @ prog.  
                        ;mem.  
  
    tablwt 0,1,0x85   ; /
```

## Writing a Block of Data

In instances where a block of `N` bytes needs to be transferred from data memory to program memory, the `tablwt` and `tlwt` instruction need to be included in a loop which checks for `N` transfers.

```
;transfer 'COUNT' bytes (even values only) of  
;data at program memory at 'RAM_BUFFER' to  
;program memory ;at 'MESSAGE'
```

```
WriteBlock  
    movlw high MESSAGE ;load table pointer  
    movpf W,TBLPTRH   ; /  
    movlw low MESSAGE ; /  
    movpf W,TBLPTRL   ; /  
    bcf ALUSTA,5      ;enable post auto  
                        ;increment of FSR0  
  
    movlw RAM_BUFFER  ;initialize FSR0 to  
                        ;RAM_BUFFER  
  
    movfp W,FSR0      ; /  
    movlw COUNT/2     ;initialize count  
  
WriteBlockLoop  
    tlwt 1,0x00       ;high byte @ table  
                        ;latch  
  
    tablwt 0,1,0x00   ;low byte @ table  
                        ;latch;  
                        ;table latch @ prog.  
                        ;mem.  
  
    decfsz W          ;check if count = 0  
    goto WriteBlockLoop ;no then do next  
    return            ;else end of transfer.
```

Author: Stan D'Souza  
Logic Products Division

# Implementing Table Read and Write in PIC17C42

---

NOTES:

---

---

# WORLDWIDE SALES & SERVICE

---

---

## AMERICAS

### Corporate Office

Microchip Technology Inc.  
2355 West Chandler Blvd.  
Chandler, AZ 85224-6199  
Tel: 602 786-7200 Fax: 602 786-7277  
Technical Support: 602 786-7627  
Web: <http://www.mchip.com/microhip>

### Atlanta

Microchip Technology Inc.  
500 Sugar Mill Road, Suite 200B  
Atlanta, GA 30350  
Tel: 770 640-0034 Fax: 770 640-0307

### Boston

Microchip Technology Inc.  
5 Mount Royal Avenue  
Marlborough, MA 01752  
Tel: 508 480-9990 Fax: 508 480-8575

### Chicago

Microchip Technology Inc.  
333 Pierce Road, Suite 180  
Itasca, IL 60143  
Tel: 708 285-0071 Fax: 708 285-0075

### Dallas

Microchip Technology Inc.  
14651 Dallas Parkway, Suite 816  
Dallas, TX 75240-8809  
Tel: 214 991-7177 Fax: 214 991-8588

### Dayton

Microchip Technology Inc.  
35 Rockridge Road  
Englewood, OH 45322  
Tel: 513 832-2543 Fax: 513 832-2841

### Los Angeles

Microchip Technology Inc.  
18201 Von Karman, Suite 455  
Irvine, CA 92715  
Tel: 714 263-1888 Fax: 714 263-1338

### New York

Microchip Technology Inc.  
150 Motor Parkway, Suite 416  
Hauppauge, NY 11788  
Tel: 516 273-5305 Fax: 516 273-5335

## AMERICAS (continued)

### San Jose

Microchip Technology Inc.  
2107 North First Street, Suite 590  
San Jose, CA 95131  
Tel: 408 436-7950 Fax: 408 436-7955

## ASIA/PACIFIC

### Hong Kong

Microchip Technology  
Unit No. 3002-3004, Tower 1  
Metroplaza  
223 Hing Fong Road  
Kwai Fong, N.T. Hong Kong  
Tel: 852 2 401 1200 Fax: 852 2 401 3431

### Korea

Microchip Technology  
168-1, Youngbo Bldg. 3 Floor  
Samsung-Dong, Kangnam-Ku,  
Seoul, Korea  
Tel: 82 2 554 7200 Fax: 82 2 558 5934

### Singapore

Microchip Technology  
200 Middle Road  
#10-03 Prime Centre  
Singapore 188980  
Tel: 65 334 8870 Fax: 65 334 8850

### Taiwan

Microchip Technology  
10F-1C 207  
Tung Hua North Road  
Taipei, Taiwan, ROC  
Tel: 886 2 717 7175 Fax: 886 2 545 0139

## EUROPE

### United Kingdom

Arizona Microchip Technology Ltd.  
Unit 6, The Courtyard  
Meadow Bank, Furlong Road  
Bourne End, Buckinghamshire SL8 5AJ  
Tel: 44 0 1628 851077 Fax: 44 0 1628 850259

### France

Arizona Microchip Technology SARL  
2 Rue du Buisson aux Fraises  
91300 Massy - France  
Tel: 33 1 69 53 63 20 Fax: 33 1 69 30 90 79

### Germany

Arizona Microchip Technology GmbH  
Gustav-Heinemann-Ring 125  
D-81739 Muenchen, Germany  
Tel: 49 89 627 144 0 Fax: 49 89 627 144 44

### Italy

Arizona Microchip Technology SRL  
Centro Direzionale Colleoni  
Palazzo Pegaso Ingresso No. 2  
Via Paracelso 23, 20041  
Agrate Brianza (MI) Italy  
Tel: 39 039 689 9939 Fax: 39 039 689 9883

## JAPAN

Microchip Technology Intl. Inc.  
Benex S-1 6F  
3-18-20, Shin Yokohama  
Kohoku-Ku, Yokohama  
Kanagawa 222 Japan  
Tel: 81 45 471 6166 Fax: 81 45 471 6122

9/22/95

All rights reserved. © 1995, Microchip Technology Incorporated, USA.

Information contained in this publication regarding device applications and the like is intended through suggestion only and may be superseded by updates. No representation or warranty is given and no liability is assumed by Microchip Technology Incorporated with respect to the accuracy or use of such information, or infringement of patents or other intellectual property rights arising from such use or otherwise. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any intellectual property rights. The Microchip logo and name are registered trademarks of Microchip Technology Inc. All rights reserved. All other trademarks mentioned herein are the property of their respective companies.