Internal EEPROM libraries for PIC18 processors

version 1.03

There are two libraries for processors with smaller (256 bytes) and larger (>256 bytes) internal EEPROMs – just like the official libs they replace (compiler chooses the right library for given processor based on data in appropriate definition file). Both libraries contain the same routines, only adjusted for EEPROM size. The files containing libraries are named __Lib_EEPROM_256.mcl and __Lib_EEPROM_1024.mcl, respectively.

The replacement libraries, besides the routines present in the original libraries, contain some additional ones. Here is the list:

EEPROM_read
EEPROM_write
EEPROM_wait
EEPROM_readBlk
EEPROM_writeBlk

EEPROM_read and EEPROM_write have the same syntax as the originals, except that EEPROM_write is a function rather than procedure, so it may be used in conditional statements to ensure that the write indeed was initiated. As the original, EEPROM_write waits for end of previous write (not indefinitely, though) but does not wait for end of present write. If one needs to wait for the end (which is advisable if there is no more bytes to write), one may use the EEPROM_wait function. It will also clear the 'write enable' bit (WREN) of EECON1 which cannot be cleared before the end of write (unlike in PIC16 processors).

Please remember that, like in the original, EEPROM_read does not wait for end of write initiated by preceding it EEPROM_write, so it is improper to call the former just after the latter. Again, EEPROM_wait may be used in between (or a delay >10ms). For any case, EEPROM_read also clears EECON1.WREN bit.

Block read/write routines have been added to allow operation on blocks of data. One may read up to 256 bytes with EEPROM_readBlk to a RAM memory locations starting at an address specified by a pointer. One should be careful not to overwrite more memory than one intended to ;).

The amount of data that can be stored in EEPROM with EEPROM_writeBlk function has been intentionally limited to 32 bytes, because writing to data EEPROM is time consuming. For example, writing 256 bytes in one go would take more than one second.

Unlike the EEPROM_write function, the EEPROM_writeBlk does wait for the end of write and clears WREN bit afterwards. It will, however, return with an error if the processor has not finished any previous writing. Therefore, it is not advisable to run EEPROM_writeBlk immediately after EEPROM_write - those two methods of writing should not be mixed, or EEPROM_wait should always follow EEPROM_write.

Examples of block writes/reads:

Code:

```
x:=1.23456;
if not EEPROM_writeBlk(SizeOf(x),$20,@x) then …
...
EEPROM_readBlk(SizeOf(x),$20,@x);


st:='EEPROM contents';
if EEPROM_writeBlk(StrLen(st)+1,$10,@st) then …
...
EEPROM_readBlk(SizeOf(st),$10,@st);
```

Note, that one doesn't have to write whole memory reserved for the string, simply the actual string plus the termination char (0 marking it's end), but one should either read byte by byte checking for the end of string, or read all space the string could take (hence SizeOf in the example) in one operation.

Library installation:

- find mP PRO installation directory and subdirectory **.../Uses/P18**,
- find original library files, **__Lib_EEPROM_256.mcl** and **__Lib_EEPROM_1024.mcl**, there and rename them,
- unpack the replacement libs and move the files to the **.../Uses/P18** directory,

The libs archive contains also an example code for PIC18F4620 processor on EasyPIC3 board with GLCD.

Have fun,

janni

EEPROM library for PIC18 processors
version 1.03
date 18.09.09
Revision history:
1.00 - modified for mP PRO 3.00
1.01 – minor change
1.02 – compiled with v4.80 beta
1.03 – compiled with v5.00

Prototypes of routines contained in both libraries are the same:

function EEPROM_read(addr:word): byte;
reads a byte from internal EEPROM at addr (clears WREN bit)

function EEPROM_write(addr:word; datab:byte):boolean;
writes data byte to specified address without waiting for end
result: true when OK
false - timeout while waiting for end of previous write

function EEPROM_wait:boolean;
waits up to 10ms for end of EEPROM write & clears WREN bit
result: true when OK
false - timeout while waiting for end of previous write

procedure EEPROM_readBlk(numb:byte; addr:word; buffer_ptr:^byte);
reads up to 256 bytes to buffer
parameters: # of bytes in numb (0 for 256 bytes), EEPROM address in addr
output: buffer - array of numb bytes

function EEPROM_writeBlk(numb:byte; addr:word; buffer_ptr:^byte):boolean;
writes up to 32 bytes from buffer to internal EEPROM
parameters: # of bytes in numb, address in addr, array of numb bytes
result: true when OK
false - timeout or write in progress, or wrong # of bytes