MMBasic For Windows

Being, in the main, the translations and transcribings from the stone tablets of one Peter Mather :)

This document produced and maintained by Mixtel90 as Divine punishment for his sins upon this earth. :)

PRE-ALPHA RELEASES

25/01/2022
Things to note:
You need to click in the window to enable the keyboard before use and after changing mode
The keyboard handling is very primitive - no auto-repeat and single keymap (UKish)
Use SHIFT-CTRL-X to exit the program tidily
Use LIST COMMANDS and LIST FUNCTIONS to see what is (should be) working
On first run the program stores a file ".options" in "MyDocuments" and the FILES command will show all files in my documents (FILES command is still very primitive - no parameters, no sort, and no paging
USE OPTION DEFAULT PATH "directory" to select the directory from where on startup you want to be able to load programs. This must be a fully specified filepath e.g. OPTION DEFAULT PATH "C:/USERS/PETER/DOCUMENTS/MMB4W". The filepath must exist.
By default the program opens with a 1024x768 window
Use OPTION DEFAULT MODE n to select the graphics mode to use on startup. The modes are as per CMM2 but always ARGB8888
Use the MODE command to switch graphics without rebooting
Use OPTION DEFAULT FONT n [,scale] to select the font to use on startup (same syntax as the font command)
The editor is the same as MM2 etc.
The LOAD "fname" [,R] and RUN "fname" both work
SAVE "fname" is not yet implemented
AUTOSAVE works but only as a way of inputting from the keyboard. Pasting into the window doesn't work and will probably never work but remember you are running on a full blown computer so you can just LOAD a file

I'll post versions here as I think useful. Don't expect any discipline - proper versioning etc.

TIME$, DATE$, EPOCH, DATETIME$, and DAY$ functions added.

NB: TIME$ and DATE$ commands will not be included as time is derived from the system clock and we don't want to be messing around with that

---

26/01
Lots more stuff implemented : text, longstring, settick, maths, sort

I've also massively reduced the CPU load. Very little now at the command prompt and just one fully active thread when a Basic program is running. The screen update rate is reduced to about 63Hz which means on a typical 60Hz refresh monitor there will be no impact. If you have a 120MHz monitor then every two frames may be the same but I think this is a price worth paying for reducing the CPU load.

NOW with full GUI and Mouse support

You can interrogate the mouse with the mouse function
MOUSE(X) 'gets the x-coordinate
MOUSE(Y) 'gets the y-coordinate
MOUSE(W [,scale]) get the movement of the wheel since the last call of the function
Use scale to adjust the sensitivity by dividing the internal count by scale (default 1)
MOUSE(L) 'gets the left mouse button (1 is pressed)
MOUSE(M) 'gets the middle mouse button (1 is pressed)
MOUSE(R) 'gets the right mouse button (1 is pressed)
MOUSE(D)
D This allows you to detect a double click of the left mouse button .
The algorithm say the two clicks must occur between 100 and 500
milliseconds apart. The report via MOUSE(D) is then valid for 500mSec
before it times out or until it is read.

NB: MM+ TOUCH() function is automatically translated to MOUSE() by the firmware, see the PicoMite manual for the touch GUI sub-functions

mouse command
MOUSE LEFTDOWNinterrupt [,RIGHTDOWNINTERRUPT] [,LEFTUPINTERRUPT]

e.g.

```
MOUSE leftint, rightint, leftup
do
 if mouse(D) then print "Double click @ ",mouse(x),mouse(y)
loop
sub leftint
 print "left down @ ",mouse(x),mouse(y)
end sub
sub leftup
 print "left up @ ",mouse(x),mouse(y)
end sub
sub rightint
 print "right down @ ",mouse(x),mouse(y)
end sub
```

OPTION MOUSE is a CMM2 command. This is windows. The mouse is always there

---
28/01
Timer function now accurate to 0.1uS and reported as a floating point number of milliseconds. Settick and pause remain as-is and have an accuracy of around +/- 16mSec. I could fix this but only by having a thread running continuously reading the high resolution timer. If you need accurate timing you can have the tight loop in basic with the TIMER function

```
timer=0:do:b=timer:?b-a:a=b:loop
```

Some other reported things fixed

29/01
I've had to rewind the work on ARGB as I got into a real mess. The attached is a tidy version of the previous with the new timer and various bug fixes. I don't see an edit issue but if you do you need to let me know how to create it

---
Caps lock should now work and I've fixed a colour error in the GUI commands plus all the prep work for the Page command and transparency is now done. This was a big job because in the old code -1 = &HFFFFFFFF which was a coded value for the text command whereas in ARGB &HFFFFFFFF is non-transparent white. This meant all colour values in the entire code needed changing to 64 bit integers from 32 and there are hundreds of them. The CMM2 didn't have this issue as there are only 15 transparency levels so &HFFFFFFF (7 F's) isn't the same as &HFFFFFFFF (8 F's)

30/01
New version pa5

Still no keyboard language support but most of the infrastructure is now there.
This version will automatically re-initialise the options on first run as the structure has changed.

Now working:
keyboard repeat
on key (both variants - see PicoMite manual)
option list
Caps lock (properly I hope)
F1 now set to FILES, F5-F9 are user settable as per PicoMite
KEYDOWN function as per CMM2

To set the keyboard repeat use OPTION KEYBOARD lang, repeatstart, repeatrate
The rates are in milliseconds. The language is ignored but is currently one of
"US", "FR", "GR", "IT", "BE", "UK", "ES"

Remember:
Use OPTION DEFAULT MODE to select the screen format on running. This automatically chooses a sensible font for the screen size. You can't select the small screen sizes which make editing silly

Use OPTION DEFAULT FONT number, scale to select the font on running. This overrides the automatic selection caused by DEFAULT MODE

Valid Default modes are:

Default mode 8", "800x600"
Default mode 8", "640x480"
Default mode 9", "1024x768"
Default mode 10", "848x480"
Default mode 11", "1280x720"
Default mode 12", "960x540"
Default mode 15", "1280x1024"
Default mode 16", "1920x1080"
Default mode 18", "1024x600"

31/01
pa6

This should support UK, US, FR and DE (GR) keyboards
Please report any success and/or issues
I have an issue with the ^ character on FR and DE keyboards as it doesn't seem to produce any recognisable keycode. Until I fix this:
on the German keyboard you can use SHIFT-3
on the French keyboard you can use SHIFT-$

New version with more improvements to keyboard/language.

US is tested with onscreen keyboard and works perfectly for me - please could someone confirm with a real keyboard

For France and Germany accent keys (e.g. circumflex) need double click as per normal use to give a single character. This needed a time-delay to get working. If you have a slower PC and it doesn't work let me know and I'll increase the timer

Note back single quote will always show as a degree-symbol as this is what is set in the fonts

UPDATE 15:00 UTC Swedish keyboard added

---
pa9

Hopefully fixes the last bits on the FR and DE keyboards
You can now run MMbasic with the name of a program to load and run

e.g. from a DOS BOX

mmbasic benchmark

will run the program "benchmark.bas". If the program doesn't exist then MMBasic will just start normally at the command prompt

---
pa10

You can now paste into autosave using ctrl-V
You can also paste into the editor but only a single line will be accepted

---

01/02
All colours are now ARGB8888. If you use the direct hex code then you must set the A.
The RGB function can take 1, 3, or 4 parameters. In the case of 3 parameters it assumes A=255 and of course defined colours like "RED" have A set to 255

---

pa11

Most file handling commands and function are now there and work based on very cursory testing. Use LIST COMMANDS and LIST FUNCTIONS to see which

The big thing missing is the DIR$ function and sorting out the FILES command. Unfortunately, windows doesn't support the "dirent.h" header I used in the Raspberry Pi so they will need coding from scratch using windows calls.

---
pa12

Last update for a few days

lots of good file stuff

FILES command and DIR$ as per the PicoMite manual
mm.info(default path)
mm.info(filesize fname$)
mkdir
rmdir

tested with ZMIM with minor tweaks. zmim directory must be in default path
dim ss$(5) 'omit the length

Sub main()
Local i. old_dir$, state, s$
ss$(0)=mm.info(default path)+"zmim"

06/02
Lots more stuff

MEMORY command, FRAMEBUFFER command, PAGE command, IMAGE command (rotate, resize, warp etc), LOAD JPG, LOAD BMP all as per CMM2 except jpg is more flexible and can overlap the screen

---

08/02
pa14

Full sprite engine and blit from CMM2 now available. Framebuffer closed on change of mode. Bug in closing files fixed

---

11/02
pa15

Starting on serial. First step to identify com ports available
Use:
LIST COM PORTS

to get a list of all ports available

use function

COMPORT(portno)

to check if a particular port exists

e.g
? COMPORT(22)

12/02
pa16

Support for serial comms (any or all of com1 - com63)
Fix for SAVE IMAGE
Fix for list ports if no ports available

Parameters available for serial comms - syntax same as all other MMbasic

Baudrate as specified, default if not 9600

Receive Buffer size as specified, default if not 256

Receive Interrupt can be specified - default no interrupt

Receive Interrupt count can be specified - default 1

Parity - default none
Available: EVEN, ODD, MARK, SPACE

Data Bits - default 8
Available 7BIT

Stop bits - default 1
Available S2, S1P5 (1.5bits)

INV not implemented
RS485 not implemented

NOTE serial output is currently blocking and the time taken will depend on the baudrate

---

pa17

Includes the 3d engine from the CMM2. See video
Fixes bug where the page command caused subsequent edits to not show the status line

pa18 ?

13/02
pa19

Should fix the framebuffer issue

---

pa20

Editor replaced with the CMM2 editor. Definite pre-alpha status

Not implemented yet:
mouse support
F7
Shift-DEL
Shift-TAB
F12
First time after starting you must specify the filename to be edited e.g
edit "myfile.bas"

Afterwards in the session it will remember the last file edited if saved (not if you esc) and you can just use
edit

use OLDEDIT to use the original MM2 editor

14/02
F7, shift-del, shift-tab now all working in the editor (see CMM2 manual for details), just mouse support to complete

typing edit without a parameter will open the currently loaded file or give an error if nothing loaded. To create a new program simply type EDIT "newfilename" and the editor will open with a blank screen

For those who haven't used the CMM2 editor before, the big advantage is that the new editor supports horizontal scrolling allowing lines longer than the screen size to be edited properly

---

pa21

Mouse now integrated into the editor with same functionality as CMM2 e.g navigating and selecting text for cut-and-paste.
Cut and paste in the editor now uses the Window clipboard so you can paste external text into the editor or get text from the editor into an external application.
Ctrl-V no longer functional in autosave - but will return soon

---

15/02
pa22

Enables double press of home and end in editor to move to start and end of the file

AUTOSAVE fname$

The filename is now mandatory as on the CMM2. You can now paste into autosave using CTRL-V. When you exit autosave with F1 or CTRL-Z. the file is saved and the program is loaded ready to run. The "last file edited" is updated so you can edit the file without specifying the filename.

In addition if you exit with F2 the file is immediately run

If you exit with CTRL-C no file will be created

LOAD DATA now implemented but completely untested.

16/02
Fix for the various drawing issues caused by an over-enthusiastic bulk edit

---

17/02
I've put a true BREAK into the firmware. WIN32 API has a routines specifically for starting and stopping the break condition. I've set the break at 20 bit periods

ALPHA RELEASES

16/02/2022
MMBasic for Windows V5.07.03a0

The MMBasic for Windows code is now officially past the pre-alpha phase and enters alpha. i.e. most of the functionality is now there and there are not too many reports of issues but there is still work to do.

The big change in V5.07.03a0 is support for proper transparency as per the CMM2

The mode command is changed as follows to follow the CMM2 as closely as possible:

MODE modeno [,alphaenabled] [,background colour]

By default alphaenabled is 0 and the background colour is BLACK

To use the second layer set alphaenabled to 1 and optionally choose a background colour

Unlike the CMM2 MMbasic for windows uses full ARGB8888 colour in all modes and has a transparency value (A) of 0-255

When you change the mode to alpha by default you are still writing to page 0 and it is cleared to RGB(BLACK)which is opaque. If you want to see the background colour pixels on page 0 must be written with a transparency of less than 255. Use CLS RGB(BLANK) to clear a page to completely transparent.
If you set to write to page 1 this will overlay page 0 and the background to the extent that the alpha value of the pixels on PAGE 1 are set (between 0=transparent and 255=opaque).

Note when playing with this at the command line you can get very confused so it is easier to understand what is happening with a very simple program.

```
mode 14,1,rgb(magenta) 'set into 2 layer mode with the background set to magenta
font 3
print "Click to get focus"
mouseclick
print "now we will clear page 0 to blank"
print "click to continue"
mouseclick
cls rgb(blank)
colour rgb(black),rgb(blank)
print "This text is written on page 0 with transparent backgound"
print "click to continue"
mouseclick
print "Now we will write to the top page"
print "click to continue"
mouseclick
page write 1
cls rgb(blank)
sprite loadpng 1,"apple"
sprite write 1,50,10,1
sprite transparency 1,128 'change the transparency of the sprite
sprite write 1,200,10,1
page write 0
```

```
print @(0,mm.info(fontheight)*4)"You can see the page 0 text through the right hand apple"
print
print "Click to exit and remember to click again to get focus"
mouseclick
sprite close all
mode 14
colour rgb(white),rgb(black)
end

sub mouseclick
do
loop until mouse(l)
do
loop until not mouse(l)
end sub
```

Finally a note on the RGB function.

RGB(COLOURNAME) will return a fully opaque representation of that colour
RGB(red, green, blue) will return a fully opaque representation of that colour
use RGB(red, green, blue, trans) to set a partially transparent colour

If you must, you can use numerical values directly as colours e.g CLS &H80607080

This will set the transparency to &H80, the blue level to &H60, the green level to &H70 and the red level to &H80. Note that red and blue are reversed compared to the CMM2 if a direct numerical value is used in this way i.e ABRG8888

V5.07.03a1

Forgot to mention above right clicking in the editor on the first character in the line is the same as HOME. Right clicking after the last character in a line on on the last character if the line exceeds the screen width is the same as the END key

This version includes the ability to send a BREAK "character" on the serial port.
e.g.
OPEN "COM1:9600" as #1
PRINT #1,BREAK
This will send a break which is set to be 20 bit lengths of the port held in the zero state. I've even tested it on the scope and it works

---
"I was expecting MMBasic-4-Windows to mimic the CMM2"

Not aware of any significant differences. The mode command is slightly different because there is only one colour depth in windows. On the CMM2 the different colour depths are there because of display performance issues and not because they are a preferred approach.

18/02
V5.07.03a3

Fixes fonts 1 and 4 output of chars above ascii 127
Fixes bug in sprites overlapping screen edge
Gets rid of duplicate IRETURN
LIST command updated to support
LIST FILE fname$
LIST FILE ALL fname$
LIST PAGES

BIG CHANGE - SOUND NOW SUPPORTED

PLAY TONE
PLAY MODFILE
PLAY MODSAMPLE
PLAY FLAC
PLAY WAV
PLAY MP3
PLAY SOUND
PLAY PAUSE
PLAY RESUME
PLAY STOP
PLAY VOLUME

Differences from CMM2:
Playing all the music files in a directory not currently supported (trivial to do with Basic)
WAV, FLAC and MP3 playback only support 44100Hz stereo
PLAY TTS not implemented
PLAY EFFECT not implemented

Please note the sound system sets up yet another thread with a callback every 1/88200 seconds so let me know if you see performance issues starting up etc.

19/02

Should fix the focus issue when changing mode
EXECUTE command implemented

Quote
Can you answer my question from above please.

I haven't implemented PLAY EFFECT yet (possibly not at all as it was always clunky. The best way to do this is to edit the effect into a MOD file and use PLAY MODSAMPLE much more efficient

---

21/02
V5.07.03pa5

I'll probably forget stuff as there are a lot of changes

Bug fixes to GUI and MID as above
Various Bug fixes to other graphics relating to transparency

New command CONSOLE. This is the same as print but puts output to the console window - perfect for debugging

Main window now has minimise and maximise capability - thanks to Romeo for the how

All # commands as per CMM2 (INCLUDE, DEFINE, COMMENT)

Mode command - use a negative value for the mode and you will get fullscreen with no border e.g. MODE -16 will fill a 1080p monitor perfectly

Mode command - now accepts 8,12, 16, and 32 as valid bits/pixel as well as 0 and 1. 8,16, and 32 are converted to 0 i.e. RGB888. 12 is converted to 1 i.e. ARGB8888

Mode 14 = Mode 12 but 2 screen pixels per logical pixel

PLAY EFFECT now implemented as per CMM2

This has all been tested with Mauro's Wolf3d and DemoX. Both run as downloaded from https://github.com/mauroxavierneto with no changes.
Gauntlet nearly runs but Mauro does some PEEK and POKEing into the framebuffer which needs changing form 8-bit to 32-bit

Issues: Still a problem with focus after mode changes - you need to click the mouse to activate keyboard input

Source open and available on https://github.com/UKTailwind/MMB4W

APPENDIX 1 - COLOURS & COLOUR BAR DEMO - By Tassy Jim

Current colours are:
WHITE YELLOW LILAC BROWN FUCHSIA RUST MAGENTA RED CYAN GREEN CERULEAN MIDGREEN COBALT MYRTLE BLUE BLACK GRAY GREY LIGHTGRAY LIGHTGREY ORANGE PINK GOLD SALMON BEIGE

```
MODE 1
CLS

TEXT 10,10 , "WHITE     "+HEX$(RGB(WHITE),4), ,1,1,RGB(WHITE)
TEXT 10,30 , "YELLOW    "+HEX$(RGB(YELLOW),4), ,1,1,RGB(YELLOW)
TEXT 10,50 , "LILAC     "+HEX$(RGB(LILAC),4), ,1,1,RGB(LILAC)
TEXT 10,70 , "BROWN     "+HEX$(RGB(BROWN),4), ,1,1,RGB(BROWN)
TEXT 10,90 , "FUCHSIA   "+HEX$(RGB(FUCHSIA),4), ,1,1,RGB(FUCHSIA)
TEXT 10,110, "RUST      "+HEX$(RGB(RUST),4), ,1,1,RGB(RUST)
TEXT 10,130, "MAGENTA   "+HEX$(RGB(MAGENTA),4), ,1,1,RGB(MAGENTA)
TEXT 10,150, "RED       "+HEX$(RGB(RED),4), ,1,1,RGB(RED)
TEXT 10,170, "CYAN      "+HEX$(RGB(CYAN),4), ,1,1,RGB(CYAN)
TEXT 10,190, "GREEN     "+HEX$(RGB(GREEN),4), ,1,1,RGB(GREEN)
TEXT 10,210, "CERULEAN  "+HEX$(RGB(CERULEAN),4), ,1,1,RGB(CERULEAN)
TEXT 10,230, "MIDGREEN  "+HEX$(RGB(MIDGREEN),4), ,1,1,RGB(MIDGREEN)
TEXT 10,250, "COBALT    "+HEX$(RGB(COBALT),4), ,1,1,RGB(COBALT)
TEXT 10,270, "MYRTLE    "+HEX$(RGB(MYRTLE),4), ,1,1,RGB(WHITE)
TEXT 10,290, "BLUE      "+HEX$(RGB(BLUE),4), ,1,1,RGB(BLUE)
TEXT 310,10 , "BLACK    "+HEX$(RGB(BLACK),4), ,1,1,RGB(WHITE)
TEXT 310,30 , "GRAY     "+HEX$(RGB(GRAY),4), ,1,1,RGB(GRAY)
TEXT 310,50 , "GREY     "+HEX$(RGB(GREY),4), ,1,1,RGB(GREY)
TEXT 310,70 , "LIGHTGRAY "+HEX$(RGB(LIGHTGRAY),4), ,1,1,RGB(LIGHTGRAY)
TEXT 310,90 , "LIGHTGREY "+HEX$(RGB(LIGHTGREY),4), ,1,1,RGB(LIGHTGREY)
TEXT 310,110, "ORANGE    "+HEX$(RGB(ORANGE),4), ,1,1,RGB(ORANGE)
TEXT 310,130, "PINK      "+HEX$(RGB(PINK),4), ,1,1,RGB(PINK)
TEXT 310,150, "GOLD      "+HEX$(RGB(GOLD),4), ,1,1,RGB(GOLD)
```

```
TEXT 310,170, "SALMON    "+HEX$(RGB(SALMON),4), ,1,1,RGB(SALMON)
TEXT 310,190, "BEIGE     "+HEX$(RGB(BEIGE),4), ,1,1,RGB(BEIGE)

BOX 180,  5,100,20,1,RGB(WHITE),RGB(WHITE)
BOX 180, 25,100,20,1,RGB(YELLOW),RGB(YELLOW)
BOX 180, 45,100,20,1,RGB(LILAC),RGB(LILAC)
BOX 180, 65,100,20,1,RGB(BROWN),RGB(BROWN)
BOX 180, 85,100,20,1,RGB(FUCHSIA),RGB(FUCHSIA)
BOX 180,105,100,20,1,RGB(RUST),RGB(RUST)
BOX 180,125,100,20,1,RGB(MAGENTA),RGB(MAGENTA)
BOX 180,145,100,20,1,RGB(RED),RGB(RED)
BOX 180,165,100,20,1,RGB(CYAN),RGB(CYAN)
BOX 180,185,100,20,1,RGB(GREEN),RGB(GREEN)
BOX 180,205,100,20,1,RGB(CERULEAN),RGB(CERULEAN)
BOX 180,225,100,20,1,RGB(MIDGREEN),RGB(MIDGREEN)
BOX 180,245,100,20,1,RGB(COBALT), RGB(COBALT)
BOX 180,265,100,20,1,RGB(MYRTLE),RGB(MYRTLE)
BOX 180,285,100,20,1,RGB(BLUE),RGB(BLUE)
BOX 480,  5,100,20,1,RGB(BLACK),RGB(BLACK)
BOX 480, 25,100,20,1,RGB(GRAY),RGB(GRAY)
BOX 480, 45,100,20,1,RGB(GREY),  RGB(GREY)
BOX 480, 65,100,20,1,RGB(LIGHTGRAY),RGB(LIGHTGRAY)
BOX 480, 85,100,20,1,RGB(LIGHTGREY),RGB(LIGHTGREY)
BOX 480,105,100,20,1,RGB(ORANGE),RGB(ORANGE)
BOX 480,125,100,20,1,RGB(PINK),RGB(PINK)
BOX 480,145,100,20,1,RGB(GOLD),RGB(GOLD)
BOX 480,165,100,20,1,RGB(SALMON),RGB(SALMON)
BOX 480,185,100,20,1,RGB(BEIGE),RGB(BEIGE)

DO:LOOP
```

Hex numbers are transparency, blue, green, red

APPENDIX 2 - SERIAL PORTS IN LINUX / WINE - By Volhout

How to use serial ports in Linux when running MMB4W under Wine.
Tested on Ubuntu 20.04LTS (20.04.03).

- Install Wine (instructions from Ubuntu website, (this does not install the latest version of Wine, but that is not essential for this post).

Most of us will use USB serial convertors using FTDI or CH340 chips. Good news: Linux does not require you to install specific drivers. All drivers (also for cheap chinese copies) are in the linux kernel.

Wine emulates windows in linux, and for serial ports it uses symbolic links to the linux serial ports. In linux these serial ports (devices) are visible as objects in the folder /dev/. The serial ports have a group identification (tty) extended with a type identification (i.e. S) and number (0...32).

The first serial port on a linux system would be /dev/ttyS0.
If you open a terminal in linux and type ls /dev/tty* you get an overview of all serial ports.

The type designator (the "S") can have different types
S = hardware serial port
USB = USB-serial convertor (this type is used by FTDI chips)
ACM = USB modem (this type is often used by Arduino and the CH340 uses it also)

Linux automatically enumerates new serial ports when a USB-serial cable is plugged in. This is a convenience (easy) and an annoyance (2 USB serial ports are not always enumerated in the same order, so you can easily accidentally swap them).

Wine

Wine lives within the Linux system in the users space. In Ubuntu the user has a home folder, and in this folder you find folders like "Documents", "Downloads", etc.. very similar to Windows user space.

In this user space Wine uses a hidden folder called ".wine" (in Linux folders becomme hidden when the folder name starts with a period "."). You can view this folder in the file browser when enabling view of hidden folders (and then you will see many more hidden folders, but .wine is one of them once you have started wine once.

In user space the folder .wine/dosdevices shows the locations where Wine interfaces with the linux system. You can find a link to your C drive, and links to serial ports.
Every time you start Wine, the folder .wine/dosdevices is refreshed, so it is not advised to make changes to the links in this folder, they will get lost.

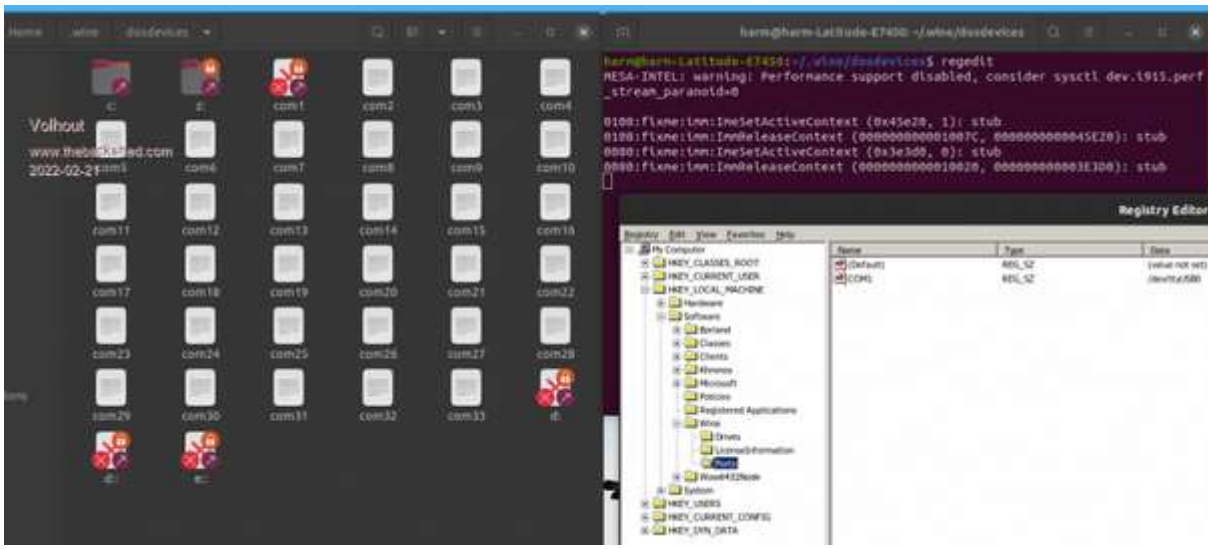
Wine re-writes the contents of this folder with links to serial ports in the /dev/ttyS* list, and these are defaulted to hardware serial ports (ttyS0...ttyS32).
It links COM1 to /dev/ttyS0, COM2 to /dev/ttyS1 etc...).

So how do we attach a USB serial port to MMB4W running under Wine...? All are attached to non existing hardware ports. Wine will add the /dev/ttyUSB0 after the first 32 hardware serial ports. so it will become COM33. When your program cannot handle COM33 (I expect MMB4W will, but some programs can not, they are restricted to COM1..COM4) then you need to force Wine to use the USB serial interface as COM1. This is how:

Regedit !

In the linux terminal, type regedit. You will start Wine's registry editor.
In HKEY_LOCAL_MACHINE/software/wine/ports you can find serial ports. This section is empty (only the default is there). When you add "COM1" with value "/dev/ttyUSB0" it will be replacing the default com1 link to /dev/ttyS0 to /dev/ttyUSB0.

See below picture that shows (in Ubuntu 20.04) the folder .win/dosdevices with many serial port links, and a link fopr the C drive. It shows the terminal where you start regeedit, and it shows an example in regedit, where I define the com1 link to /dev/ttyUSB0



For basic usage, when only 1 or 2 USB-serial interfaces are attached, this may suffice. You simply add a second entry in Regedit with com2 linking to /dev/ttyUSB1

For more complex setups, Linux can use rules, based on the manufacurer and serial number of the USB-serial convertor chip to identify each of them uniquely. When there is interest I can write a post about that.

So does it work .... YES

U used following simple terminal program to communicate between MMB4W running under Wine, with a CMM2. In between a FTDI TTL-232RG cable.

```
'very simple terminal program

open "com1:9600" as #1

do
 do
  if loc(#1) <> 0 then
    b$=input$(loc(#1),#1)
    print b$;
  end if
  a$=inkey$
 loop until a$<>""
 print #1,a$;
loop until a$="q"
```

The only quirck I found is that the CMM2 and MMB4W under Wine use a different terminator (<LF> and <CR><LF>) but that is easily overcome....