

MAXIDASH dashboard and IDE

[maxidash_1b.zip](#) [maxidash_1c.zip](#) [maxidash.zip](#)

MAXIDASH is a dashboard, and an IDE of sorts, for the (colour) maximate.

MAXIDASH is a MMBASIC 4 program that runs on the MM to experiment with an MM and to build event-driven programs. You can interactively control pins/ports and enter values and formulas for pins, pretty much like you would enter values and formulas into a spreadsheet. The formulas can be saved and exported to a stand-alone .BAS program.

MAXIDASH.BAS for the CMM and MM

Pin modes can be set interactively

Values are continuously updated and displayed (here at 31 FPS)

Values can be interactively entered as constants or as formulas

(formula for P22/D1 shown)

Syntax hints for entering formulas

Current selection is P22/D1

Memory bank with 10 variables, their values entered as constants or as formulas

Timer T (ms)

Dashboard refresh rate

Auto-save and export as .BAS program

PIN	J9	MODE	VALUE	PIN	D/A	MODE	VALUE	MEM	MODE	VALUE
1	C21	----		21	D0	DOUT:	0	M0	HOLD:	1000
2	C23	----		22	D1	DOUT:	0	M1	HOLD:	0
3	C25	----		23	D2	DIN:	1	M2	HOLD:	0
4	C27	----		24	D3	----		M3	HOLD:	0
5	C29	----		25	D4	----		M4	HOLD:	0
6	C31	----		26	D5	----		M5	HOLD:	0
7	C33	----		27	D6	----		M6	HOLD:	0
8	C35	----		28	D7	----		M7	HOLD:	0
9	C37	----		29	D8	----		M8	HOLD:	0
10	C39	----		30	D9	----		M9	HOLD:	0
11	C4	----		31	D10	----		PWM CH1:		
12	C6	----		32	D11	----		PWM CH2:		
13	C8	----		33	D12	----		PIN0LED:		
14	C10	----		34	D13	----				
15	C12	----		35	A0	AIN:	0.287097			
16	C14	----		36	A1	AIN:	0.265591	T:	30282	
17	C16	----		37	A2	AIN:	0.289247	FPS:	31	
18	C18	----		38	A3	AIN:	0.270968	AUTO SAVE:	ZZZZDASH	
19	C20	----		39	A4	AIN:	0.267742			
20	C22	----		40	A5	AIN:	0.248387			

= <t\m0>%2

PIN/MEM commands:	Program commands:	Navigation commands:	
I set input PIN	S save	UP/DOWN	M R.
O set output PIN	E export	TAB/LEFT/RIGHT	A A.
- disable PIN	= assign formula	PAGEUP/PAGEDOWN	X v.
SPACE (un)hold value	ENTER edit formula	HOME/END	I E
			D n
			A g
			S e
			H l
			e
			1a n
			2018

Formula operators: | Formula functions: | Formula operands: |

| ? : if-then-else | ABS ATN COS DEG EXP | P0-P40 PIN |

| || && conditionals | FIX INT INV LOG MAX | M0-M9 MEM |

| | ^ & bit-wise | MIN POW RAD RND SGN | C4-C22 J9 connector |

| != < <= > >= | SHL SHR SIN SQR TAN | D0-D13 and A0-A5 |

| + - * / \ % ~ ! | HIGH/LOW/KEY/T/numbr |

The export feature currently saves the exported program as a .DSH file which is still an MMBASIC program, since the program itself saves as a .BAS file using the specified file name (ZZZZDASH by default). I'm currently using CHAIN to extend the program when formulas are entered by the user, but perhaps LIBRARY LOAD can be used instead to speed this up in the future.

Download MAXIDASH by selecting the attachments (upper right corner).

MAXIDASH is easy to use and can be easily extended by adding new functions you want to use in formulas, by extending the functions\$ string in MAXIDASH.BAS and BASEDASH.BAS. For example, to add support for DS18B20(n), add "1:DS18B20," to functions\$, where 1 means one argument is required.

Below are simple MAXIDASH examples. These simple examples should give you a very basic idea how

MAXIDASH is programmed. The first is just a simple blinking LED (pin0 is switched each second using the timer value specified as T):

```
PIN0LED = (T\1000) % 2
```

The second example lights the LED when the voltage on A0 drops below 1.5V. First select A0 then change its mode to AIN by typing 'I' and use the following formula for PIN0LED:

```
PIN0LED = A0 < 1.5
```

The third example runs 8 LEDs on D0-D7 as a continuous wave of 4 LEDs lit up:

```
M0 = 1000
M1 = 125
M2 = 500
D0 = t % m0 < m2
D1 = (t+m1) % m0 < m2
D2 = (t+2*m1) % m0 < m2
D3 = (t+3*m1) % m0 < m2
D4 = (t+4*m1) % m0 < m2
D5 = (t+5*m1) % m0 < m2
D6 = (t+6*m1) % m0 < m2
D7 = (t+7*m1) % m0 < m2
```

The variables M0, M1 and M2 can be interactively changed in MAXIDASH to influence the pattern, or you can create formulas for M0-M2 based on the timer value (t) or a key press (key) value, e.g. M2=key?10*key:M2 will set M2 to the ASCII value times 10 when a key is pressed.

For example, the following simple example sets D0 to 0 when key '0' is pressed and to 1 when key '1' is pressed, and toggles D1 and D2 with the 'a' and 'b' keys, respectively:

```
D0 = key=48 ? 0 : key=49 ? 1 : D0
D1 = key=97
D2 = key=98
```

Now, when you connect a 74HC595 shift register pin DS to D0, pin ST_CP to D1 and SH_CP to D2 you can now interactively control the shift register from the keyboard! Just type '0'/'1' to set bit value, shift one bit in with key 'a' and latch with key 'b'. Export the program as ZZZZDASH.DSH (or rename) and you can run ZZZZDASH.DSH as a fast standalone program or integrate it in a project.

It is also easy to shift values into the 74HC595 using a counter M2 of the "memory bank" that is initially set to M1=16 (for 16 half cycles) and to let M3 shift a bit into the register each cycle, starting with the initial value of M0. M0 is the value we want to shift in, for example &haa:

```
M0 = &haa
M1 = 16
M2 = M1 ? M1 : M2 ? M2-1 : 0
M3 = M1 ? M0 : M2%2 ? M3 : SHR(M3, 1)
```

The 74HC595 shift register pin DS connects to D0, pin ST_CP to D1 and SH_CP to D2:

```
D0 = M3%2
D1 = M1 ? 0 : M2 ? M2%2 : 0
D2 = key=98
```

When you interactively set M1 to 0 then the M2 counter will count down, triggering D1 to flip each cycle, and triggering M3 to shift from low to high bits into the register via D0. Then press 'b' on the keyboard to move the value to the register's memory. Alternatively, assign

```
D2 = M2=0
```

to rise the SH_CP edge after the bits shifted in.

From:

<http://fruitoftheshed.com/wiki/> - FotS

Permanent link:

http://fruitoftheshed.com/wiki/doku.php?id=mmbasic:maxidash_dashboard_and_ide

Last update: **2024/01/19 09:30**

