

Care and feeding your new Colour Maximite 2.

There are some things that need setting up to enable you to get the best from your new toy.

On first power up, you should have set the keyboard but this can be achieved at any time with

```
OPTION USBKEYBOARD UK | US | DE | FR
```

The default is SDcard timing is:

```
OPTION SD TIMING NORMAL
```

You may want to try:

```
OPTION SD TIMING FAST
```

If you are running firmware prior to V5.05.04 and having trouble reading your SDcards, try setting the option:

```
OPTION SD TIMING CONSERVATIVE
```

That is equivalent to the NORMAL setting in later firmware.

Adjusting the monitor:

You can test all the graphic modes to test your monitor. Most monitors will need some adjustment to get the best out of all resolutions.

You should also note the best aspect ratio for your monitor. This will be handy later if you want circles appear round.

A suitable program to cycle through all modes is:

```
' test card for CMM2
' TassyJim May 2020
OPTION EXPLICIT
OPTION DEFAULT NONE
DIM INTEGER wd, ht, wbox, sh, x, w, n, nn, m
DIM FLOAT a
DIM k$, imgtitle$, fname$
DIM INTEGER c(8)
c(0) = RGB(BLACK)
c(1) = RGB(YELLOW)
c(2) = RGB(CYAN)
c(3) = RGB(GREEN)
c(4) = RGB(MAGENTA)
c(5) = RGB(RED)
c(6) = RGB(BLUE)
c(7) = RGB(WHITE)
c(8) = RGB(64,64,64)
CLS

DO
  SELECT CASE m
    CASE 0
      MODE 1, 8
    CASE 1
      MODE 1, 8
      nn = 10
      a = 1 ' aspect ratio used in the CIRCLE command.
      imgtitle$ = " MODE 1,8 Ratio "+STR$(a,1,2)+" "
    CASE 2
```

```

MODE 1, 12
nn = 10
a = 1 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 1,12 Ratio "+STR$(a,1,2)+" "
CASE 3
MODE 1, 16
nn = 10
a = 1 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 1,16 Ratio "+STR$(a,1,2)+" "
CASE 4
MODE 2, 8
nn = 8
a = 1.08 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 2,8 Ratio "+STR$(a,1,2)+" "
CASE 5
MODE 2,12
nn = 8
a = 1.08 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 2,12 Ratio "+STR$(a,1,2)+" "
CASE 6
MODE 2,16
nn = 8
a = 1.08 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 2,16 Ratio "+STR$(a,1,2)+" "
CASE 7
MODE 3, 8
nn = 4
a = 1.08 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 3,8 Ratio "+STR$(a,1,2)+" "
CASE 8
MODE 3,12
nn = 4
a = 1.08 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 3,12 Ratio "+STR$(a,1,2)+" "
CASE 9
MODE 3,16
nn = 4
a = 1.08 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 3,16 Ratio "+STR$(a,1,2)+" "
CASE 10
MODE 4, 8
nn = 4
a = 0.833 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 4,8 Ratio "+STR$(a,1,3)+" "
CASE 11
MODE 4,12
nn = 4
a = 0.833 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 4,12 Ratio "+STR$(a,1,3)+" "
CASE 12
MODE 4,16
nn = 4
a = 0.833 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 4,16 Ratio "+STR$(a,1,3)+" "
CASE 13
MODE 5, 8
nn = 4
a = 0.833 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 5,8 Ratio "+STR$(a,1,3)+" "
CASE 14
MODE 5,12
nn = 4
a = 0.833 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 5,12 Ratio "+STR$(a,1,3)+" "
CASE 15
MODE 5,16
nn = 4
a = 0.833 ' aspect ratio used in the CIRCLE command.
imgtitle$ = " MODE 5,16 Ratio "+STR$(a,1,3)+" "

END SELECT
'PRINT imgtitle$ ' for debug
IF m = 0 THEN
CLS
TEXT 400,100, "Video mode test",cm,5,1
TEXT 400,180, "Ratio = aspect ratio used in the circle command",cm,1,1
TEXT 400,220, "Q to quit, P to save page as a BMP",cm,3,1
TEXT 400,260,"UP arrow to go back, any other key to advance",cm,3,1
ELSE
wd = MM.HRES : ht = MM.VRES
wbox = wd / 8
FOR x = 0 TO 7
BOX x*wbox,ht/4,wbox,ht/2,0,c(x), c(x)
NEXT x
FOR x = 0 TO wd-1
sh = 255*x/wd

```

```

LINE x,0,x,ht/12,1,RGB(sh,0,0)
LINE x,ht/12,x,ht/6,1,RGB(0,sh,0)
LINE x,ht/6,x,ht/4,1,RGB(0,0,sh)

LINE x,ht*9/12,x,ht*10/12,1,RGB(0,sh,sh)
LINE x,ht*10/12,x,ht*11/12,1,RGB(sh,0,sh)
LINE x,ht*11/12,x,ht,1,RGB(sh,sh,0)

LINE x,ht/2,x,ht*3/4,1,RGB(sh,sh,sh)
NEXT x
CIRCLE wd/2,ht/2, ht*15/32,3,a,c(7)
sh = 0
x = wd/2 - 55*nn/2
FOR w = 10 TO 1 STEP -1
FOR n = 1 TO nn
sh = 255 - sh
LINE x,ht*3/8,x,ht*5/8,w,RGB(sh,sh,sh)
x = x + w
NEXT n
NEXT w

BOX 0,0,wd,ht,3,c(7)
BOX 1,1,wd-2,ht-2,1,c(5)
TEXT wd/2,ht/2, imgtitle$,cm,1,1
ENDIF

DO
k$ = INKEY$
LOOP UNTIL k$<>" "

IF k$ = "P" OR k$ = "p" THEN
fname$ = MID$(imgtitle$,2)+".bmp"
SAVE IMAGE fname$
TEXT wd/2,ht/2, "Saved as "+fname$,cm,1,1
DO
k$ = INKEY$
LOOP UNTIL k$<>" "
ENDIF
IF k$ = "Q" OR k$ = "q" THEN EXIT DO
'PRINT ASC(k$)
IF ASC(k$) = 128 THEN
m = m - 1
IF m < 1 THEN m = 15
ELSE
m = m + 1
IF m > 15 THEN m = 1
ENDIF
ENDIF

LOOP
MODE 1,8 ' restore before ending program
CLS

```

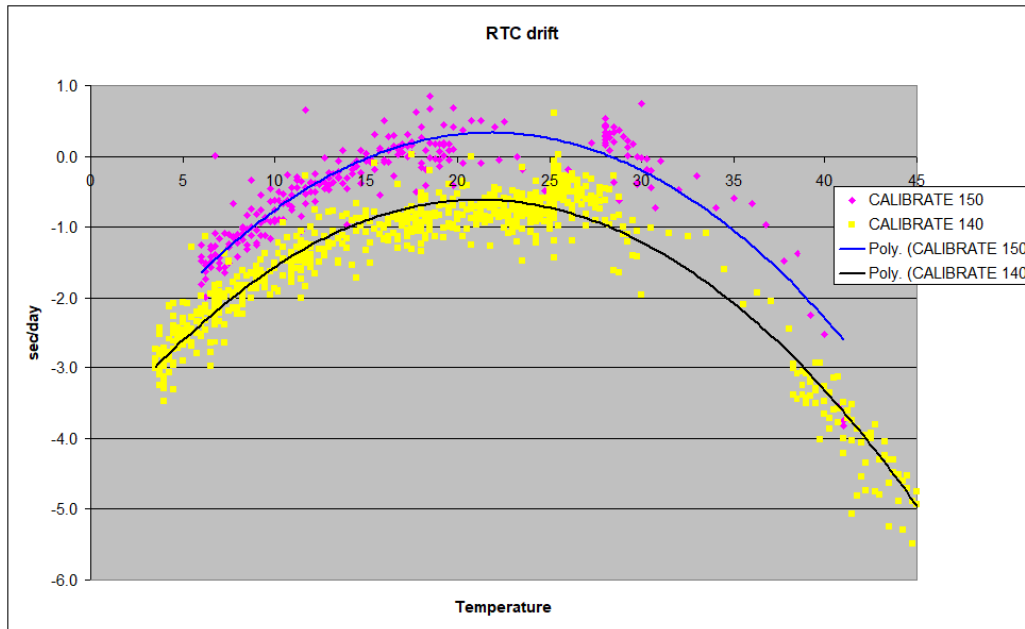
Adjusting the clock speed.

The RTC used a standard watch crystal on the Waveshare module and a surface mount crystal on the all-in-one board. Both systems will follow the typical frequency curve for ‘tuning fork’ crystals.

The RTC has adjustment available and once set, keeps good time. It can be set to better than 1ppm.

The CMM2 adjusts the effective frequency by adding or dropping pulses every 32 seconds. $32768 \times 32 = 1024$, giving us the 1ppm resolution available.

The frequency does vary with temperature and every board will require a different setting. One system I have gave this:



As a comparison the DS3231/DS3232 RTCs have a specified accuracy of 2 ppm. Not as good within the “room temperature” range but it does have the advantage of a much flatter temperature curve.

You can set the RTC by trial and error or a quicker way, if you have a GPS with one second pulses. (Most of the GPS modules do this)

Connect the pps output to PIN 38 and run the following program.

The program retrieves the current RTC CALIBRATE setting and then compares the RTC with the GPS every 60 seconds.

It also checks the main oscillator but there is no adjustment available for it, just the RTC.

Every 16 minutes, the program adjusts the RTC CALIBRATE setting.

```
' clock drift test and RTC CALIBRATE setting
' TassyJim June 2020
'
DIM INTEGER calValue = 0          ' initial calibrate vale
DIM GPSint = 38                  ' connect to GPS lpps pulse
DIM interval = 60                 ' time between calculations in seconds
DIM adjustTime = 16              ' number of intervals to use for setting RTC CALIBRATE
DIM perDay = 60*60*24/interval   ' calculations per day
DIM FLOAT cal = 0.0824           ' secs per day for each unit of RTC CALIBRATE
DIM INTEGER tick, tock, longCount
```

```

DIM GPStime$
DIM FLOAT caseTemp
DIM FLOAT RTctime, lastRTctime, OSCtime, lastOSCtime
DIM FLOAT RTCinterval, OSCinterval, OSCerror
DIM FLOAT firstRTC, firstOSC, longRTC, longOSC
DIM FLOAT RTCperDay
DIM INTEGER longTime = LEN(TIMES) ' check to see if MILLISECONDS ON is set"
OPTION milliseconds ON
CLS
calValue = PEEK(BYTE &h58004070) + PEEK(BYTE &h58004071)*256 ' retrieve old setting
IF calValue > 512 THEN calValue = calValue - 256*256
PRINT ""
PRINT "Test started at ";TIMES
PRINT "OPTION RTC CALIBRATE is set to ";calValue
PRINT
PRINT "          Time          RTC          8MHz Osc          Room "
PRINT "          drift S/Day n      Main OSC      ppm      Temp "
PRINT
SETPIN GPSint, INTH, GPStick, PULLDOWN
tick = -1
DO
  IF tock THEN
    doTock
    tock = 0
  ENDIF
LOOP UNTIL INKEY$ <> "" ' any key to exit
PAUSE 10
IF longTime = 8 THEN ' restore normal TIMES setting
  OPTION milliseconds OFF
  PRINT "Time reset to short mode"
ENDIF
PRINT TIMES
PRINT "OPTION RTC CALIBRATE is set to ";calValue

END

SUB GPStick
  tick = tick + 1
  IF (tick MOD interval) = 0 THEN
    tock = 1
    GPStime$ = TIMES
    OSCtime = TIMER
  ENDIF
END SUB

SUB doTock
  caseTemp = TEMPR(42)
  IF caseTemp > 100 THEN caseTemp = 0
  RTctime = VAL(FIELDS$(GPStime$,3,":")) ' only interested in the seconds field
  IF tick = 0 THEN
    firstRTC = RTctime
    firstOSC = OSCtime
    longCount = 0
    longRTC = 0
    longOSC = 0
    RTCinterval = 0
    OSCinterval = 0
    OSCerror = 0
  ELSE
    RTCinterval = RTctime - lastRTctime
    OSCinterval = OSCtime - lastOSCtime
    OSCerror = interval*1000 - OSCinterval
    longCount = longCount + 1
    longRTC = (RTctime - firstRTC)/longCount ' average since program start
    longOSC = (interval*1000 - (OSCtime - firstOSC)/longCount)*1000/interval ' ppm
  ENDIF
  RTCperDay = longRTC*perDay
  lastRTctime = RTctime
  lastOSCtime = OSCtime

  PRINT STR$(longcount,2,0)+" "+GPStime$;" ";STR$(RTCinterval,3,3);
  PRINT " ";STR$(RTCperDay,3,2);" ";STR$(-RTCperDay/cal,4,0);" ";
  PRINT STR$(OSCerror,3,3);" ";STR$(longOSC,3,3);" ";caseTemp

  IF (tick MOD (interval*adjustTime)) = 0 AND tick > 0 THEN ' reset the counter for adjusting
    IF ABS( RTCperDay/cal) > 1.5 THEN ' only change if there is something worth doing
      calValue = calValue -RTCperDay/cal
      OPTION RTC CALIBRATE calValue
    ENDIF
    PRINT "New RTC CALIBRATE : ";calValue
    tick = -1 ' force reset next tick
  ENDIF
END SUB

```

If you don't have a handy GPS module you can use a Windows program to adjust the RTC to your computer time.

MMTimeLord can be downloaded from <https://www.c-com.com.au/MMedit.htm>

Using the IO pins.

The maximum current you can safely draw or sink from any IO pin is 20 mA. It is preferable to keep it below 8mA to prevent pulling the voltage too far from the 3.3 or ground level.

The maximum for all pins should be limited to 100mA

If a pin is set to DIN with PULLUP enabled, there is a ~40k ohm resistor connected to 3.3V Likewise, PULLDOWN set a ~40k resistor to ground.

The pullup/pulldown resistance can vary considerably.

The changeover level between a high and a low is approximately 1.5-1.6 Volts

Digital pins are 5 Volt tolerant but if you are feeding in 5V, PULLUP or PULLDOWN should **not** be used on those pins.

The voltage seen by digital-out pins used as open collector should also be restricted to 5V.

Analogue pins are limited to 0 - 3.3V

Pin 40 is a special case.

It is used for a system reset. If pin 40 is tied to ground during start-up, the system is reset, wiping out all options and any program in memory.

MMBasic applies the PULLUP to pin 40 for 300mS during the startup test so any resistance less than ~30k ohms to ground is likely to cause a reset.

Capacitance should not be a problem and values up to 1uF have been tested.

Long cables have also been tested without any issues.

Pin 40 is often used for SPI2 CLOCK in which case there will be a pullup resistor on the attached module and this will **not** cause a reset.

A common way to use the IO port is with an RPi expansion board plugged into a breadboard.

The IDE cable used must be connected the correct way around or there will be problems.

Pin 1 on the CMM2 is on the left looking from the rear. If your IDE cable has locating lugs, the standard socket will have its cut-out on the wrong side. You can easily cut a second cut-out on the bottom of the socket, allowing the cable to be inserted with pin1 (red trace) correctly on the left.

Don't be tempted to use an 80 way IDE cable. It has ground connections internally connected and they don't match the Maximize grounds.

The audio is directly connected to the CPU and expects a load of about 5k, in line with normal audio line-in levels.

When playing audio, there is a DC offset of VCC/2 or 1.65V.

If you are feeding into PC powered speakers etc, a 4.7k resistor (and optionally 1uF capacitor) in series is suggested.

If you have any interference in the audio, you may need a “Ground Loop Isolator”
In Australia, the Jaycar AA3086 should be a suitable device.



The audio output can also be used as a DC-out for a variable DC voltage without the need for PWM and filtering. In that case, blocking capacitors should not be used.