- XEQM is a new custom menu hidden on a long press of XEQ. It is meant to be a PC configurable custom menu, which has nothing to do with the MyMenu system which still will work as per WP43S.

  The 18 soft keys of XEQM are user programmable using (currently) a very basic subset of commands. That will expand later.

  On the flash drive is a PROGRAMS directory, with any number up to 18 text script files named XEQM00.TXT through XEQM18.TXT, and these reflect the 18 softkey positions in XEQM.

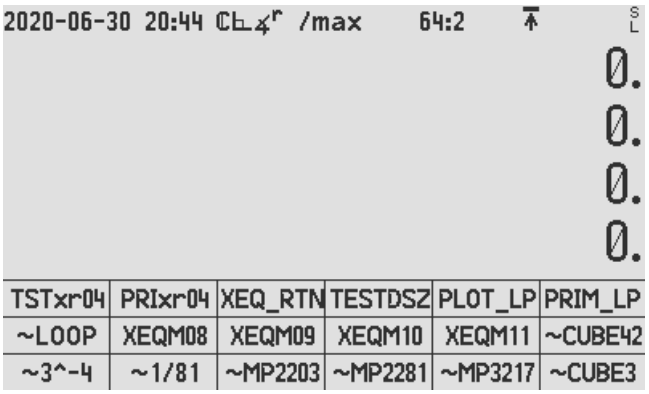

Figure 1: XEQM menu

- Long press on XEQ shows cycles from XEQM, to NOP. Release the button when XEQM is shown.

- PROGRAMS folder must be in the same folder C43 is run from, on WINDOWS.
  PROGRAMS folder must be in the root on DM42 flash drive.
  PROGRAM folder to contain text files named: XEQM01.TXT … XEQM18.TXT

- Demo programs for g[FN1] … g[FN6] are supplied in zip folder.
  Place them in the PROGRAMS directory.
  If no files found in PROGRAMS (XEQMnn.TXT), [FN1] … [FN6] are demo programs, starting with ~ are present.

- [FN1]: ~3^-4, a demo program trying to do 3^-4 using integers, which currently produces 0. This is in the process of being fixed by 43S team.

- [FN2]: ~1/81, a demo program doing `1 ENTER 81 /` and the answer is as expected.

- [FN3]: ~MP2203, a demo program calculating `2 ENTER 2203 Y^X -1 PRIME?` which is a Mersenne Prime and doing a prime check. The execution time is given in seconds.

- [FN4]: ~MP2281, a demo program calculating `2 ENTER 2281 Y^X -1 PRIME?` which is a Mersenne Prime and doing a prime check. The execution time is given in seconds.

- [FN5]: ~MP3217, a demo program calculating `2 ENTER 3217 Y^X -1 PRIME?` which is a Mersenne Prime and doing a prime check. The execution time is given in seconds.

- [FN6]: ~CUBE3, a demo program that enters the "3 cubes = 3" problem constants, 569936821221962380720, -569936821113563493509 & -472715493453327032, cubes them and adds them together to get 3.

- g[FN1]: XEQM13.TXT, **TSTxr04**,
  Demo: //PROGRAM TO PLOT 40 EMPTY LOOP CALC TIMES ON A GRAPH
  Usage: Store a multiplier 20 to Register 04 and run g[FN1].
  Example: 20 ENTER STO 04 Longpress XEQ to call up XEQM, press g[FN1].
  Result: Graph of times taken to perform the loop overhead on the 40 numbers 10^0, 10^20, 10^40, … 10^780.
  Result: xxx.STAT.TSV file in the C43 folder, with tab separated values for this data.

- g[FN2], XEQM14.TXT, **PRIxr04**,
  Demo PROGRAM TO PLOT 40 PRIME NUMBER CALC TIMES ON A GRAPH\
  Same as above, but times include the Next Prime function.
  Result: Graph of times taken to perform NEXTP (next prime) on the 40 primes following 10^0, 10^20, 10^40, … 10^780.
  Result: xxx.STAT.TSV file in the C43 folder, with tab separated values for this data.

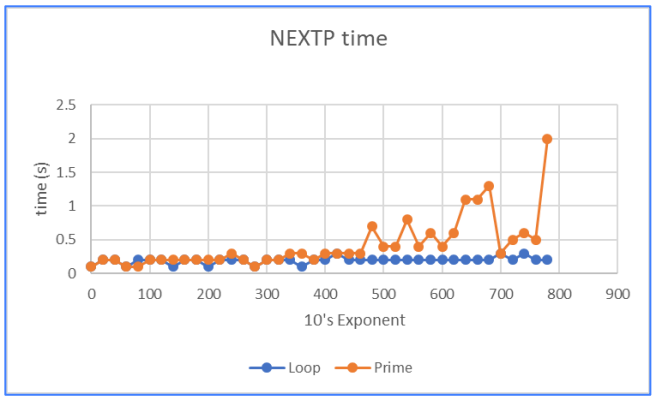

Figure 3: Example of TSTxr04 and PRIxr04 running on Windows Simulator. The resulting TSV file was loaded into Excel, and the number pairs plotted to show the time taken for the loop overhead (blue) and prime number calculations (brown).

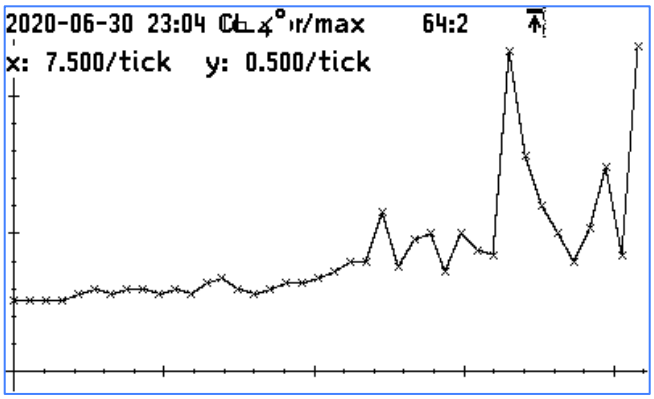

Figure 2: Another example of PRIxr04 running on the DM42. Snapshot was made of the DM42 screen, showing the time taken on the DM42 to caluclate the series of prime numbers.

- g[FN3]: XEQM15.TXT, XEQ_RTN,
  Demo //PROGRAM TO DEMO SUBROUTINES AND DSZ

- g[FN4]: XEQM16.TXT, TESTDSZ,
  Demo //PROGRAM TO DEMO DSZ AND GOTO

- g[FN5]: XEQM17.TXT, **PLOT_LP**
  Demo //PROGRAM TO PLOT RANDOM NUMBERS ON A LINEAR GRAPH

- g[FN6], XEQM18.TXT, PRIM_LP

  Demo //PROGRAM TO DEMO DSZ AND GOTO. GET FIRST 4 PRIMES AFTER 1E100 TO STACK

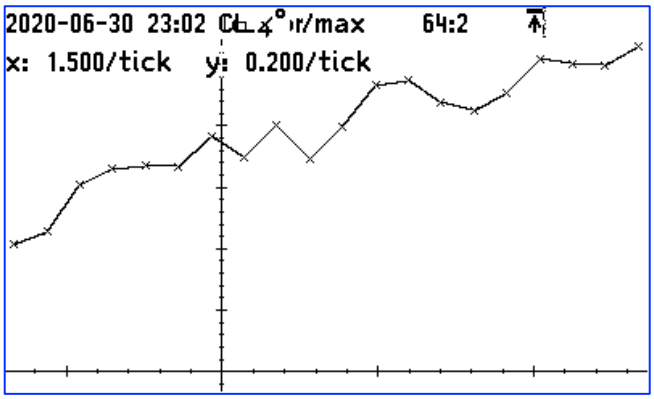

Figure 4: Example of PLOT_LP, from a snapshot off the DM42, plotting random numbers along a linear line.