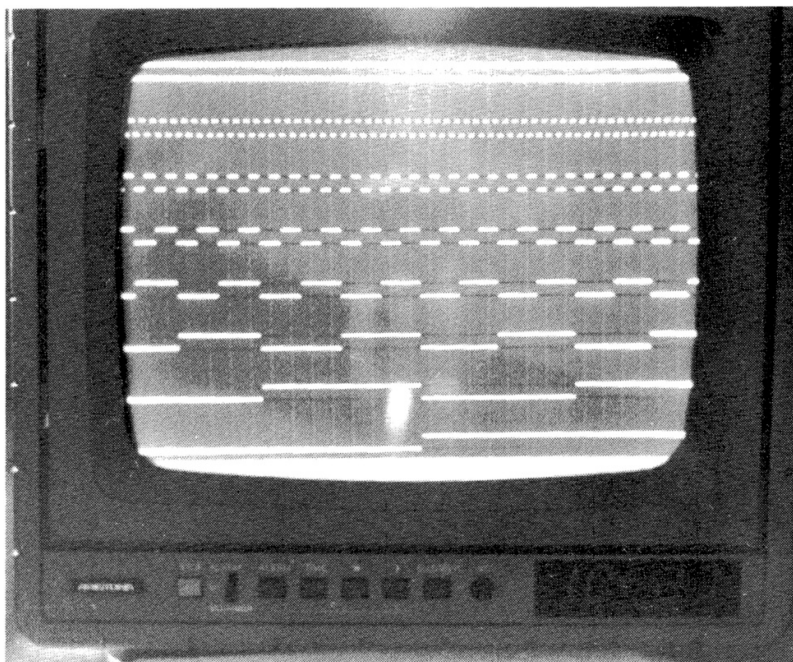




Digiscope

for 6502 systems

M. Dohmen



Those who have a Graphic Display and a computer system based on the 6502 will discover in this article that the possibilities of this combination are almost unlimited. The January 1980 issue described a frequency meter, which once again showed that a computer can prove its worth in the hobbyist's electronics lab. The program discussed here, as an addition to the frequency meter, will for many be an answer to the question: "Should I buy a computer?" What we have here is a so-called "digiscope". This is a kind of oscilloscope that makes digital signals visible on the graphic display.

This digiscope is fairly fast. Without hardware modifications the maximum measurable frequency is 71.5 kHz. It has eight channels, so eight frequencies can be measured simultaneously. With some hardware extensions a mini logic analyzer can be built. If you only want to measure frequencies and not, for example, pulse/pause ratios, I advise you to use ordinary divide-by-10 counters to extend the frequency range. A drawback of

the digiscope is that it only knows two states at its inputs, namely "high" and "low", so it is only usable in experiments with digital components. An advantage is that, thanks to the eight channels, the position of the signals relative to each other can be seen very well. The shape of the signal is also easily recognisable with this program (provided the frequency does not get too high). The program had to meet the following requirements:

1. High speed and high resolution.
2. Eight channels.
3. It must be possible to make the period time visible.

Such a program can be realised in three ways. First of all, use can be made of the processor's SO pin (sometimes also called RO). This is the fastest method. The SO pin is directly connected to the Overflow bit in the status register. By making the level on SO go from high to low, the Overflow bit is reset. Using this pin as an input makes very fast input possible. For our purpose, however, it is not suitable, because it is one input instead of eight. Here we will have to use an I/O port. A second possibility is to sample the input port at fixed intervals and then convert the relevant bits into eight dots for the graphic display. A serious drawback, however, is the limited speed. Setting a dot on the graphic display takes a lot of time (64 μ s), so the frequency range becomes far too small. By far the most effective method is to read in 256 bytes in succession, store them and then convert them into eight graphs for the graphic display. This can be implemented in two ways. Both will be discussed.

Lijst 1

```

0320 ;   INGANGSRoutine VOOR EEN MAXIMALE
0330 ;   FREQUENTIE VAN 38,5 KHZ
0340 ;
0350 ;
0200- A2 00 0360 INPUT      LDX #000      ;HAAL 256 MAAL DE
0202- A0 40 17 0370 LUS      LDA PAD2      ;WAARDE BINNEN
0205- 95 00 0380           STA *BUFFER,X    ;VAN DE PIA-POORT
0207- E8 0390           INX      ;EN ZET DEZE IN DE
0208- D0 F8 0400           BNE LUS      ;BUFFER
020A- 4C 00 07 0410           JMP DSPL     ;SPRING NAAR GO-ROUTINE

```

Lijst 1 Korte invoerroutine.

Implementation

Fetching and storing the 256 bytes can be done in a program loop; see listing 1. The advantage of this program is that it is very short. A disadvantage that weighs heavily, however, is the speed. Fetching one byte takes no less than 13 μ s. The frequency range is then only 38.5 kHz. Those who are satisfied with this can type in the program from listing 1. It uses all locations on page zero, but since the monitor is not used, this is not much of a problem.

X is the counter that indicates where the fetched byte is to be placed in page zero. The operation is now fairly easy to follow. The input port is read 256 times and the values are stored. When this has been done, a jump is made to the display routine.

The possibility now to be discussed is indeed much faster, but uses no less than 1¼K of memory. For people with a small memory it is therefore not so attractive. Now, however, the promised frequency range of 71.5 kHz is achieved. In the first method a lot of time

was wasted incrementing X, comparing with zero and the relative branch (about 5 μ s altogether). To gain this time it is necessary to write out the loop completely and not use indexed addressing. In this way 6 μ s is gained, so that the time to read in a byte and write it away is reduced to 7 μ s, which amounts to a frequency range of 71.5 kHz. The program then looks as shown in listing 2. It would be going too far to type all this in. That is what we have a computer for, which can do it faster and more accurately. A program that types in the 1¼K input program for you can be found in listing 3.

Lijst 2

```

0490 ; INKANGROUTINE VOOR EEN MAXIMALE
0495 ; FREQUENTIE VAN 71.5 KHZ
0496 ;
0497 ;
0498 ;
0499 ;
0500 ;
0501 ;
0502 ;
0503 ;
0504 ;
0505 ;
0506 ;
0507 ;
0508 ;
0509 ;
0510 ;
0511 ;
0512 ;
0513 ;
0514 ;
0515 ;
0516 ;
0517 ;
0518 ;
0519 ;
0520 ;
0521 ;
0522 ;
0523 ;
0524 ;
0525 ;
0526 ;
0527 ;
0528 ;
0529 ;
0530 ;
0531 ;
0532 ;
0533 ;
0534 ;
0535 ;
0536 ;
0537 ;
0538 ;
0539 ;
0540 ;
0541 ;
0542 ;
0543 ;
0544 ;
0545 ;
0546 ;
0547 ;
0548 ;
0549 ;
0550 ;
0551 ;
0552 ;
0553 ;
0554 ;
0555 ;
0556 ;
0557 ;
0558 ;
0559 ;
0560 ;
0561 ;
0562 ;
0563 ;
0564 ;
0565 ;
0566 ;
0567 ;
0568 ;
0569 ;
0570 ;
0571 ;
0572 ;
0573 ;
0574 ;
0575 ;
0576 ;
0577 ;
0578 ;
0579 ;
0580 ;
0581 ;
0582 ;
0583 ;
0584 ;
0585 ;
0586 ;
0587 ;
0588 ;
0589 ;
0590 ;
0591 ;
0592 ;
0593 ;
0594 ;
0595 ;
0596 ;
0597 ;
0598 ;
0599 ;
0600 ;
0601 ;
0602 ;
0603 ;
0604 ;
0605 ;
0606 ;
0607 ;
0608 ;
0609 ;
0610 ;
0611 ;
0612 ;
0613 ;
0614 ;
0615 ;
0616 ;
0617 ;
0618 ;
0619 ;
0620 ;
0621 ;
0622 ;
0623 ;
0624 ;
0625 ;
0626 ;
0627 ;
0628 ;
0629 ;
0630 ;
0631 ;
0632 ;
0633 ;
0634 ;
0635 ;
0636 ;
0637 ;
0638 ;
0639 ;
0640 ;
0641 ;
0642 ;
0643 ;
0644 ;
0645 ;
0646 ;
0647 ;
0648 ;
0649 ;
0650 ;
0651 ;
0652 ;
0653 ;
0654 ;
0655 ;
0656 ;
0657 ;
0658 ;
0659 ;
0660 ;
0661 ;
0662 ;
0663 ;
0664 ;
0665 ;
0666 ;
0667 ;
0668 ;
0669 ;
0670 ;
0671 ;
0672 ;
0673 ;
0674 ;
0675 ;
0676 ;
0677 ;
0678 ;
0679 ;
0680 ;
0681 ;
0682 ;
0683 ;
0684 ;
0685 ;
0686 ;
0687 ;
0688 ;
0689 ;
0690 ;
0691 ;
0692 ;
0693 ;
0694 ;
0695 ;
0696 ;
0697 ;
0698 ;
0699 ;
0700 ;
0701 ;
0702 ;
0703 ;
0704 ;
0705 ;
0706 ;
0707 ;
0708 ;
0709 ;
0710 ;
0711 ;
0712 ;
0713 ;
0714 ;
0715 ;
0716 ;
0717 ;
0718 ;
0719 ;
0720 ;
0721 ;
0722 ;
0723 ;
0724 ;
0725 ;
0726 ;
0727 ;
0728 ;
0729 ;
0730 ;
0731 ;
0732 ;
0733 ;
0734 ;
0735 ;
0736 ;
0737 ;
0738 ;
0739 ;
0740 ;
0741 ;
0742 ;
0743 ;
0744 ;
0745 ;
0746 ;
0747 ;
0748 ;
0749 ;
0750 ;
0751 ;
0752 ;
0753 ;
0754 ;
0755 ;
0756 ;
0757 ;
0758 ;
0759 ;
0760 ;
0761 ;
0762 ;
0763 ;
0764 ;
0765 ;
0766 ;
0767 ;
0768 ;
0769 ;
0770 ;
0771 ;
0772 ;
0773 ;
0774 ;
0775 ;
0776 ;
0777 ;
0778 ;
0779 ;
0780 ;
0781 ;
0782 ;
0783 ;
0784 ;
0785 ;
0786 ;
0787 ;
0788 ;
0789 ;
0790 ;
0791 ;
0792 ;
0793 ;
0794 ;
0795 ;
0796 ;
0797 ;
0798 ;
0799 ;
0800 ;
0801 ;
0802 ;
0803 ;
0804 ;
0805 ;
0806 ;
0807 ;
0808 ;
0809 ;
0810 ;
0811 ;
0812 ;
0813 ;
0814 ;
0815 ;
0816 ;
0817 ;
0818 ;
0819 ;
0820 ;
0821 ;
0822 ;
0823 ;
0824 ;
0825 ;
0826 ;
0827 ;
0828 ;
0829 ;
0830 ;
0831 ;
0832 ;
0833 ;
0834 ;
0835 ;
0836 ;
0837 ;
0838 ;
0839 ;
0840 ;
0841 ;
0842 ;
0843 ;
0844 ;
0845 ;
0846 ;
0847 ;
0848 ;
0849 ;
0850 ;
0851 ;
0852 ;
0853 ;
0854 ;
0855 ;
0856 ;
0857 ;
0858 ;
0859 ;
0860 ;
0861 ;
0862 ;
0863 ;
0864 ;
0865 ;
0866 ;
0867 ;
0868 ;
0869 ;
0870 ;
0871 ;
0872 ;
0873 ;
0874 ;
0875 ;
0876 ;
0877 ;
0878 ;
0879 ;
0880 ;
0881 ;
0882 ;
0883 ;
0884 ;
0885 ;
0886 ;
0887 ;
0888 ;
0889 ;
0890 ;
0891 ;
0892 ;
0893 ;
0894 ;
0895 ;
0896 ;
0897 ;
0898 ;
0899 ;
0900 ;
0901 ;
0902 ;
0903 ;
0904 ;
0905 ;
0906 ;
0907 ;
0908 ;
0909 ;
0910 ;
0911 ;
0912 ;
0913 ;
0914 ;
0915 ;
0916 ;
0917 ;
0918 ;
0919 ;
0920 ;
0921 ;
0922 ;
0923 ;
0924 ;
0925 ;
0926 ;
0927 ;
0928 ;
0929 ;
0930 ;
0931 ;
0932 ;
0933 ;
0934 ;
0935 ;
0936 ;
0937 ;
0938 ;
0939 ;
0940 ;
0941 ;
0942 ;
0943 ;
0944 ;
0945 ;
0946 ;
0947 ;
0948 ;
0949 ;
0950 ;
0951 ;
0952 ;
0953 ;
0954 ;
0955 ;
0956 ;
0957 ;
0958 ;
0959 ;
0960 ;
0961 ;
0962 ;
0963 ;
0964 ;
0965 ;
0966 ;
0967 ;
0968 ;
0969 ;
0970 ;
0971 ;
0972 ;
0973 ;
0974 ;
0975 ;
0976 ;
0977 ;
0978 ;
0979 ;
0980 ;
0981 ;
0982 ;
0983 ;
0984 ;
0985 ;
0986 ;
0987 ;
0988 ;
0989 ;
0990 ;
0991 ;
0992 ;
0993 ;
0994 ;
0995 ;
0996 ;
0997 ;
0998 ;
0999 ;

```

Lijst 3

```

1270 ; PROGRAMMA DAT DE SNELLE INLEZROUTINE
1271 ; IN HET GEHEUGEN PLACHT
1272 ;
1273 ;
1274 ;
1275 ;
1276 ;
1277 ;
1278 ;
1279 ;
1280 ;
1281 ;
1282 ;
1283 ;
1284 ;
1285 ;
1286 ;
1287 ;
1288 ;
1289 ;
1290 ;
1291 ;
1292 ;
1293 ;
1294 ;
1295 ;
1296 ;
1297 ;
1298 ;
1299 ;
1300 ;
1301 ;
1302 ;
1303 ;
1304 ;
1305 ;
1306 ;
1307 ;
1308 ;
1309 ;
1310 ;
1311 ;
1312 ;
1313 ;
1314 ;
1315 ;
1316 ;
1317 ;
1318 ;
1319 ;
1320 ;
1321 ;
1322 ;
1323 ;
1324 ;
1325 ;
1326 ;
1327 ;
1328 ;
1329 ;
1330 ;
1331 ;
1332 ;
1333 ;
1334 ;
1335 ;
1336 ;
1337 ;
1338 ;
1339 ;
1340 ;
1341 ;
1342 ;
1343 ;
1344 ;
1345 ;
1346 ;
1347 ;
1348 ;
1349 ;
1350 ;
1351 ;
1352 ;
1353 ;
1354 ;
1355 ;
1356 ;
1357 ;
1358 ;
1359 ;
1360 ;
1361 ;
1362 ;
1363 ;
1364 ;
1365 ;
1366 ;
1367 ;
1368 ;
1369 ;
1370 ;
1371 ;
1372 ;
1373 ;
1374 ;
1375 ;
1376 ;
1377 ;
1378 ;
1379 ;
1380 ;
1381 ;
1382 ;
1383 ;
1384 ;
1385 ;
1386 ;
1387 ;
1388 ;
1389 ;
1390 ;
1391 ;
1392 ;
1393 ;
1394 ;
1395 ;
1396 ;
1397 ;
1398 ;
1399 ;
1400 ;
1401 ;
1402 ;
1403 ;
1404 ;
1405 ;
1406 ;
1407 ;
1408 ;
1409 ;
1410 ;
1411 ;
1412 ;
1413 ;
1414 ;
1415 ;
1416 ;
1417 ;
1418 ;
1419 ;
1420 ;
1421 ;
1422 ;
1423 ;
1424 ;
1425 ;
1426 ;
1427 ;
1428 ;
1429 ;
1430 ;
1431 ;
1432 ;
1433 ;
1434 ;
1435 ;
1436 ;
1437 ;
1438 ;
1439 ;
1440 ;
1441 ;
1442 ;
1443 ;
1444 ;
1445 ;
1446 ;
1447 ;
1448 ;
1449 ;
1450 ;
1451 ;
1452 ;
1453 ;
1454 ;
1455 ;
1456 ;
1457 ;
1458 ;
1459 ;
1460 ;
1461 ;
1462 ;
1463 ;
1464 ;
1465 ;
1466 ;
1467 ;
1468 ;
1469 ;
1470 ;
1471 ;
1472 ;
1473 ;
1474 ;
1475 ;
1476 ;
1477 ;
1478 ;
1479 ;
1480 ;
1481 ;
1482 ;
1483 ;
1484 ;
1485 ;
1486 ;
1487 ;
1488 ;
1489 ;
1490 ;
1491 ;
1492 ;
1493 ;
1494 ;
1495 ;
1496 ;
1497 ;
1498 ;
1499 ;
1500 ;
1501 ;
1502 ;
1503 ;
1504 ;
1505 ;
1506 ;
1507 ;
1508 ;
1509 ;
1510 ;
1511 ;
1512 ;
1513 ;
1514 ;
1515 ;
1516 ;
1517 ;
1518 ;
1519 ;
1520 ;
1521 ;
1522 ;
1523 ;
1524 ;
1525 ;
1526 ;
1527 ;
1528 ;
1529 ;
1530 ;
1531 ;
1532 ;
1533 ;
1534 ;
1535 ;
1536 ;
1537 ;
1538 ;
1539 ;
1540 ;
1541 ;
1542 ;
1543 ;
1544 ;
1545 ;
1546 ;
1547 ;
1548 ;
1549 ;
1550 ;
1551 ;
1552 ;
1553 ;
1554 ;
1555 ;
1556 ;
1557 ;
1558 ;
1559 ;
1560 ;
1561 ;
1562 ;
1563 ;
1564 ;
1565 ;
1566 ;
1567 ;
1568 ;
1569 ;
1570 ;
1571 ;
1572 ;
1573 ;
1574 ;
1575 ;
1576 ;
1577 ;
1578 ;
1579 ;
1580 ;
1581 ;
1582 ;
1583 ;
1584 ;
1585 ;
1586 ;
1587 ;
1588 ;
1589 ;
1590 ;
1591 ;
1592 ;
1593 ;
1594 ;
1595 ;
1596 ;
1597 ;
1598 ;
1599 ;
1600 ;
1601 ;
1602 ;
1603 ;
1604 ;
1605 ;
1606 ;
1607 ;
1608 ;
1609 ;
1610 ;
1611 ;
1612 ;
1613 ;
1614 ;
1615 ;
1616 ;
1617 ;
1618 ;
1619 ;
1620 ;
1621 ;
1622 ;
1623 ;
1624 ;
1625 ;
1626 ;
1627 ;
1628 ;
1629 ;
1630 ;
1631 ;
1632 ;
1633 ;
1634 ;
1635 ;
1636 ;
1637 ;
1638 ;
1639 ;
1640 ;
1641 ;
1642 ;
1643 ;
1644 ;
1645 ;
1646 ;
1647 ;
1648 ;
1649 ;
1650 ;
1651 ;
1652 ;
1653 ;
1654 ;
1655 ;
1656 ;
1657 ;
1658 ;
1659 ;
1660 ;
1661 ;
1662 ;
1663 ;
1664 ;
1665 ;
1666 ;
1667 ;
1668 ;
1669 ;
1670 ;
1671 ;
1672 ;
1673 ;
1674 ;
1675 ;
1676 ;
1677 ;
1678 ;
1679 ;
1680 ;
1681 ;
1682 ;
1683 ;
1684 ;
1685 ;
1686 ;
1687 ;
1688 ;
1689 ;
1690 ;
1691 ;
1692 ;
1693 ;
1694 ;
1695 ;
1696 ;
1697 ;
1698 ;
1699 ;
1700 ;
1701 ;
1702 ;
1703 ;
1704 ;
1705 ;
1706 ;
1707 ;
1708 ;
1709 ;
1710 ;
1711 ;
1712 ;
1713 ;
1714 ;
1715 ;
1716 ;
1717 ;
1718 ;
1719 ;
1720 ;
1721 ;
1722 ;
1723 ;
1724 ;
1725 ;
1726 ;
1727 ;
1728 ;
1729 ;
1730 ;
1731 ;
1732 ;
1733 ;
1734 ;
1735 ;
1736 ;
1737 ;
1738 ;
1739 ;
1740 ;
1741 ;
1742 ;
1743 ;
1744 ;
1745 ;
1746 ;
1747 ;
1748 ;
1749 ;
1750 ;
1751 ;
1752 ;
1753 ;
1754 ;
1755 ;
1756 ;
1757 ;
1758 ;
1759 ;
1760 ;
1761 ;
1762 ;
1763 ;
1764 ;
1765 ;
1766 ;
1767 ;
1768 ;
1769 ;
1770 ;
1771 ;
1772 ;
1773 ;
1774 ;
1775 ;
1776 ;
1777 ;
1778 ;
1779 ;
1780 ;
1781 ;
1782 ;
1783 ;
1784 ;
1785 ;
1786 ;
1787 ;
1788 ;
1789 ;
1790 ;
1791 ;
1792 ;
1793 ;
1794 ;
1795 ;
1796 ;
1797 ;
1798 ;
1799 ;
1800 ;
1801 ;
1802 ;
1803 ;
1804 ;
1805 ;
1806 ;
1807 ;
1808 ;
1809 ;
1810 ;
1811 ;
1812 ;
1813 ;
1814 ;
1815 ;
1816 ;
1817 ;
1818 ;
1819 ;
1820 ;
1821 ;
1822 ;
1823 ;
1824 ;
1825 ;
1826 ;
1827 ;
1828 ;
1829 ;
1830 ;
1831 ;
1832 ;
1833 ;
1834 ;
1835 ;
1836 ;
1837 ;
1838 ;
1839 ;
1840 ;
1841 ;
1842 ;
1843 ;
1844 ;
1845 ;
1846 ;
1847 ;
1848 ;
1849 ;
1850 ;
1851 ;
1852 ;
1853 ;
1854 ;
1855 ;
1856 ;
1857 ;
1858 ;
1859 ;
1860 ;
1861 ;
1862 ;
1863 ;
1864 ;
1865 ;
1866 ;
1867 ;
1868 ;
1869 ;
1870 ;
1871 ;
1872 ;
1873 ;
1874 ;
1875 ;
1876 ;
1877 ;
1878 ;
1879 ;
1880 ;
1881 ;
1882 ;
1883 ;
1884 ;
1885 ;
1886 ;
1887 ;
1888 ;
1889 ;
1890 ;
1891 ;
1892 ;
1893 ;
1894 ;
1895 ;
1896 ;
1897 ;
1898 ;
1899 ;
1900 ;
1901 ;
1902 ;
1903 ;
1904 ;
1905 ;
1906 ;
1907 ;
1908 ;
1909 ;
1910 ;
1911 ;
1912 ;
1913 ;
1914 ;
1915 ;
1916 ;
1917 ;
1918 ;
1919 ;
1920 ;
1921 ;
1922 ;
1923 ;
1924 ;
1925 ;
1926 ;
1927 ;
1928 ;
1929 ;
1930 ;
1931 ;
1932 ;
1933 ;
1934 ;
1935 ;
1936 ;
1937 ;
1938 ;
1939 ;
1940 ;
1941 ;
1942 ;
1943 ;
1944 ;
1945 ;
1946 ;
1947 ;
1948 ;
1949 ;
1950 ;
1951 ;
1952 ;
1953 ;
1954 ;
1955 ;
1956 ;
1957 ;
1958 ;
1959 ;
1960 ;
1961 ;
1962 ;
1963 ;
1964 ;
1965 ;
1966 ;
1967 ;
1968 ;
1969 ;
1970 ;
1971 ;
1972 ;
1973 ;
1974 ;
1975 ;
1976 ;
1977 ;
1978 ;
1979 ;
1980 ;
1981 ;
1982 ;
1983 ;
1984 ;
1985 ;
1986 ;
1987 ;
1988 ;
1989 ;
1990 ;
1991 ;
1992 ;
1993 ;
1994 ;
1995 ;
1996 ;
1997 ;
1998 ;
1999 ;

```

Operation

The program for fast input, which must be placed consecutively in memory, is in table 3 in listing 3. Register X counts from 0 to 255. Two locations on page 0 keep track of where the table is to be "printed". In this case from \$0200 to \$0700. At \$0700 the display routine begins, which converts the fetched values into eight graphs. The table is four bytes long and contains: AD 40 17 85.

The fifth byte is the location on page 0 where the byte fetched from port A is to be stored. This fifth byte, however, must run from \$00 to \$FF. Register X takes care of this. Y is loaded with \$04. X is transferred to the accumulator and the fifth byte is placed in the relevant memory location. Y is decremented by one and then the four values from the table are placed one after another in the memory area. Now X is incremented by one and the pointer by five (\$0783). Once this has happened 256 times, the memory area is filled and a jump is made to \$0200, where the program just written starts. If in your case the display routine cannot be placed directly after the input routine, the boxed part of listing 3 must be inserted. This ensures that a jump to the display routine is placed after the input routine. It takes three bytes from table 4 of this listing and puts them in memory starting at \$0700. Of course the correct start address must then be entered in table 4. If your display routine starts at \$1000, for example, table 4 must become: 4C 00 10. Note the order, low byte first.

Display routine

No page 0 locations can be used for the display routine, which makes it somewhat longer. The program sets eight dots for each X coordinate. The Y coordinate of those dots is determined on the one hand by the channel, and on the other hand by the logic level of that channel. For this purpose the Y coordinates for the respective channels have been put in a table, namely a separate table for the "1" levels of the eight channels (table 1 in listing 4) and one for the "0" levels (table 2). In this way each channel can be placed anywhere on the screen. This is handy when text also has to appear on the screen.

The input routine has made sure that the fetched values have been placed in consecutive locations on page 0. Register X is now used both as the pointer referring to the addresses on page 0 and as the X coordinate of the dot on the graphic display. With this knowledge the program in listing 4 becomes somewhat easier to understand. First the I/O registers are set correctly and the screen is cleared. Then X is loaded with \$00 (X coordinate is 0, pointer points to location \$00) and Y with \$07, since eight dots must be set per byte. By shifting one bit of the byte into the carry each time, each bit is converted into a dot for the graphic display. For this purpose Y is saved at location \$0769. The first byte is fetched, shifted one place to the right and stored back. If the carry is "1", a high level must be written; if it is "0", a low level must be written. To do this, entry Y from table 1 or entry Y from table 2 must be put into Y itself. This is realised as follows. The program assumes that the carry is "0" and therefore a value is fetched from table 2 (\$0728), depending on the value of Y. The instructions STA, LDA and LDY respectively do not affect the carry bit, so the carry still determines whether a high or a low level must be drawn. At address \$072B it is now stated that the next instruction (fetch the value from table 2) must be skipped if the carry bit equals "1". On arriving at address \$0731, the X coordinate of the dot is therefore in X and the Y coordinate in Y. Then a jump is made to the routine that sets a white dot at these coordinates. The same happens for the seven other channels, after which X is incremented and a new value is fetched from page 0. After 2048 dots have been set, a jump back to the monitor is made.

Finishing touches

When everything has been typed in, the program can be tested. Check especially the "input-routine-writing" part (\$076A onwards), because this routine writes into RAM and there is a great risk that you will lose all your programs if an error has crept in while typing. Listing 5 shows a test program that gives the result shown in fig. 1 if everything works. If that is not the case, something is wrong.

[illegible]

```

      1640 2          TESTROUTINE
      1650 2
      1660 2
      1670          .BR #0000
      1680 2
      1690 2
0000-02 00          LDR #0000          ;SCHARJF BUFFER
0002-0A          STA JZUL NET
0003-95 00          TFR +BUFFER,X
0005-E5          RAR 2          ;NULLEN
0006-D0 FA          BRF LUS 1
0008-7C 00 07       JMP DPL1          ;SPRING NAHR DISPLAYROUTINE

```

```

1790      .LN #0000
1800      ;
1810      ;
1820      ;
1830      ;
1840      ;
1850      ;
1860      MEET
1870      LUSR
1880      CB
1890
1900-20 FF 07 1890
1910-08 FB 1900
1920-05 00 1910 LUSB
1930-08 FB 1920
1940-20 FF 07 1930
1950-08 FB 1940
1960-08 01 1950
1970-08 00 1960 LUSC
1980-08 FB 1970
1990-08 1C 1980
2000-20 FF 07 1990
2010-08 F7 2000
2020-05 00 2010 LUSD
2030-08 FB 2020
2040-08 1C 2030
2050-20 FF 07 2040
2060-08 F7 2050
2070-08 98 2060
2080-08 69 07 2070
2090-08 0A 2080
2100-08 0A 2090
2110-08 0A 2100
2120-08 39 2110
2130-08 ED 69 07 2120
2140-4C 1D 1C 2130
2150      LVA #000
2160      WACHT OP
2170      #BUFER.X
2180      MEEN
2190      INX
2200      #OPRANDE
2210      INX
2220      #FLANK
2230      BNE LUSX ;
2240      LVA #BUFER.X
2250      WACHT OP
2260      SUE EERSTE
2270      INX
2280      #OPRANDE
2290      BEG LUSX
2300      #FLANK
2310      STELLER = 1
2320      WACHT OP
2330      LVA #BUFER.X
2340      INX
2350      #VOLGENDE
2360      INX
2370      #OPRANDE
2380      INX
2390      #FLANK
2400      BNE LUSX ;
2410      LVA #BUFER.X
2420      WACHT OP
2430      INX
2440      #OPRANDE
2450      INX
2460      #FLANK
2470      BEG LUSX
2480      MEET HANTAL
2490      TWA
2500      #AFRIJDEN
2510      STA VSXU
2520      #BEWAAR DIT
2530      #VERWENTGAARDIG
2540      RSL A
2550      MEET ?
2560      RSL A
2570      MEET TREK ER
2580      SEC
2590      #V VORAF
2600      SEC VSXU ;
2610      JNF MONIT

```

Because of the way the program is set up, it is fairly easy to display, for example, period times on the graphic display. With the aid of the program "Letters on the graphic display" by D. M. de Boer (RB, May and July 1979) you can add text and numbers. Listing 6 shows the program for measuring the period time of a signal. The principle is as follows. Fetch a byte, select a bit and count how long it takes before the bit has gone from "0" to "1" and from "1" back to "0". First it must be found when the bit becomes "0". From there on counting must be done. Register X is the pointer that selects the relevant byte. Address 07FF must contain which channel is selected, in the form $2^{(n-1)}$, where n is the relevant channel. So if channel 3 is to be selected, \$04 must be put in \$07FF. When the program shown in listing 6 is run, address 0700 must contain: 4C0008. This is necessary because the display routine erases the measured values and no period time could be measured any more.

First of all X is made equal to zero. Then a byte is fetched from page 0 and the relevant bit is masked. If this bit is "1", the next bit must be fetched. After all, counting only starts at a transition from "0" to "1". X is incremented each time until the bit has become "0". Then X is incremented until the bit becomes "1" again. We now know for certain that we are at the start of a new period. Y is loaded with "1" (we have already had one cycle) and X and Y are incremented each time until the relevant bit goes from "0" to "1". Y now contains the period time divided by seven, since each dot represents 7 μ s. The value in

Y still has to be multiplied by seven. This is done as follows. Y is saved in \$0801 and also transferred to the accumulator. The accumulator is shifted left three times (i.e. multiplied by eight), after which Y is subtracted once. The result is: $Y * (8 - 1)$ in the accumulator. The value in the accumulator can now be put on the screen with the program "Letters on the graphic display".

Listing 1 Short input routine.

Listing 2 Long but fast input routine.

Listing 3 Program that writes the fast input routine into memory.

Listing 4 Routine for displaying logic signals on the graphic display.

Listing 5 Test program.

Listing 6 Program for measuring period times.