D.A.R.K.

Single-chip microcomputer controlled
multi-channel data collection system

by

Jørgen Franck

DANSK AMATØR RAKET KLUB

Single-chip microcomputer controlled multi-channel data collection system.

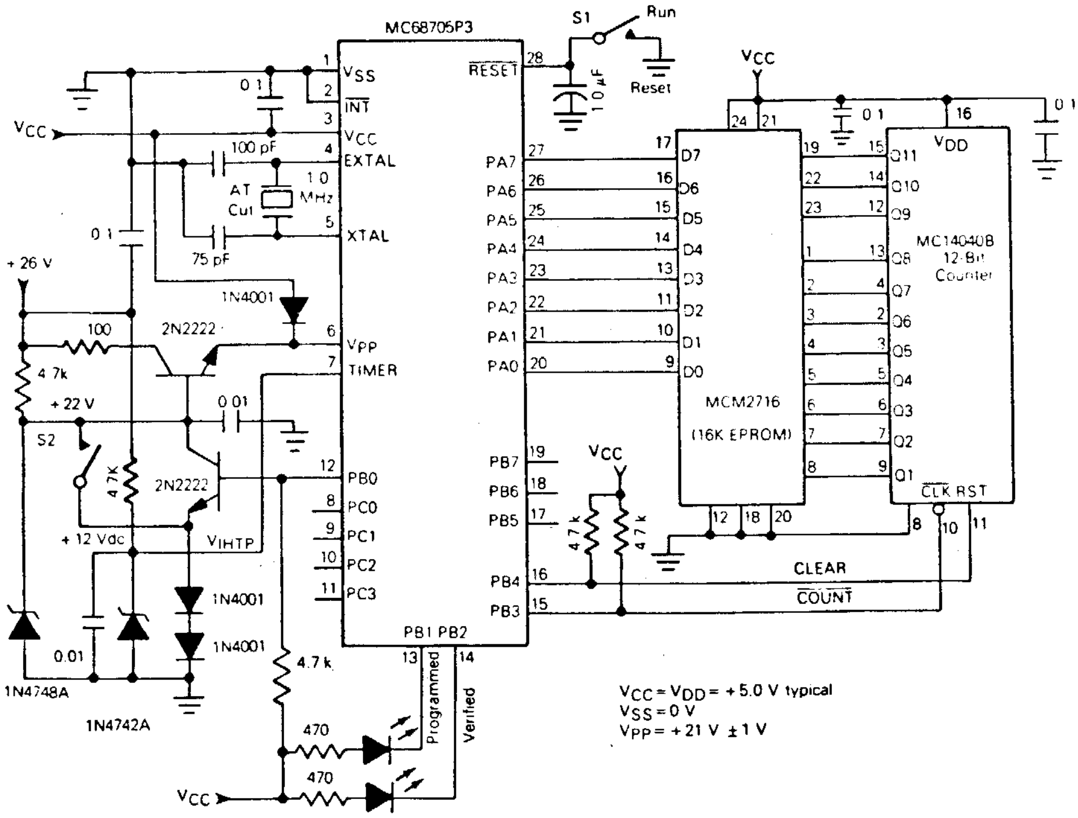
Jøgen Franck, DARK.

With a single-chip microcomputer controlled data collection system you get a larger flexibility. The sequence in which the channels are scanned, and the scanning rate of each channel, can be programmed separately for each flight. Further the single-chip microcomputer can perform other tasks than controlling the data collection. An other advantage is the simpel print layout.

One of the first problems you face, is how to program the single-chip microcomputer. Normally you can buy very expensive development kits to the different single-chip microcomputers on the market.
To avoid buying an expensive development kit, we have chosen a MOTOROLA MC68705P3 single-chip microcomputer. It is a member of the 6805 family of microprocessors with a pruned 6800 cpu. Extra instructions are included to improve its performance in control applications and it has 1.8 Kbyte of EPROM, 112 bytes of RAM and 20 pins that can be individually programmed as inputs or outputs.

A special feature of the MC68705P3 is that it contains a bootstrap program in ROM which is used by the processor to co-ordinate a self-programming procedure. Figure 1 provides a schematic diagram of the programming circuit.

First you program an ordinary EPROM (2716) and then you insert it in the programming circuit, the processor will then automatically transfer code from tne external EPROM to its own EPROM. Programming is also followed by a verification check.



Summary of Programming Steps:

1 When plugging in the MC68705P3 or the MCM2716, be sure that S1 and S2 are closed and that V_{CC} and + 26 V are not applied

2 To initiate programming, be sure S1 is closed, S2 is closed, and V_{CC} and + 26 V are applied. Then open S2, followed by S1

3 Before removing the MC68705P3, first close S2 and then close S1. Disconnect V_{CC} and + 26 V, then remove the MC68705P3

Figure 1. MC68705P3 programming interface circuit diagram.

So you need a EPROM-programmer before you can program the MC68705P3. First you have to build a I/O port to control the EPROM-programmer. As I/O chip we use the INTEL 8255A programmable peripheral interface. The schematic circuit diagram over the I/O port for a APPLE II microcomputer is shown in figure 2. Finally the schematic circuit diagram over the EPROM-programmer is shown in figure 3.

You can also simulate the EPROM and 12-bit counter (fig. 1) with a microcomputer. Then you simply feed the databits into the MC68705P3's EPROM from the I/O port. You must then use a flip-flop fore the handshake signals $\overline{\text{COUNT}}$ and CLEAR, to be sure that you catch them.

If you place the I/O port in slot nr. 3, the following address allocations is established.

Address	I/O port
49328	port A
49329	port B
49330	port C
49331	control word

If you want more information about the I/O port connections to the APPLE II microcomputer, I can refer to another article in this conference book. about analog-to-digital converters for the APPLE II.

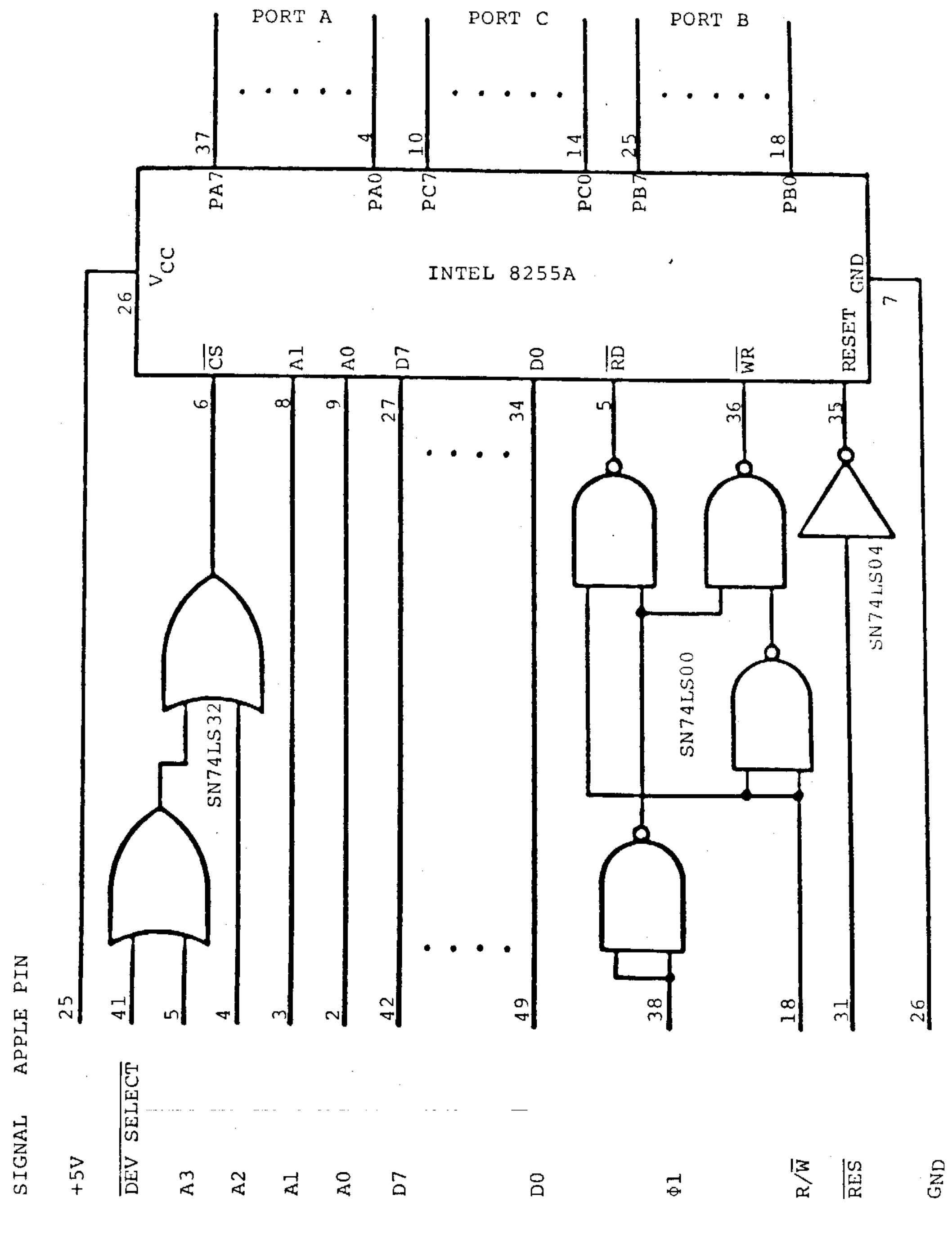


Figure 2. I/O port interface circuit schematic.

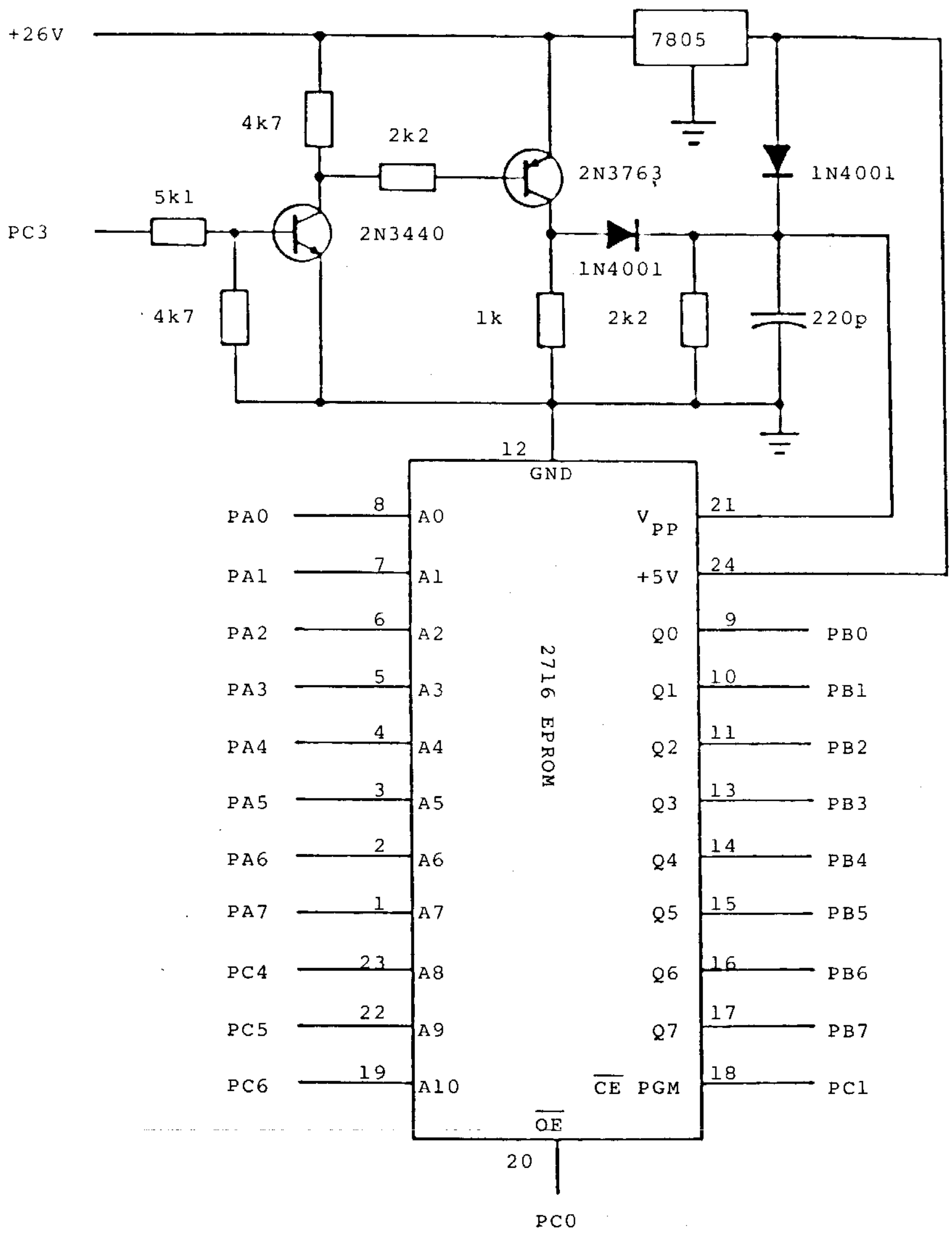


Figure 3. 2716 EPROM-prog. interface circuit diagram.

The following BASIC program can read the contents of the 2716 EPROM.

```
10  REM  READ 2716 EPROM
20 X =  - 1:A = 0
30  POKE 49331,130
40  FOR I = 0 TO 255
50 X = X + 1
60  POKE 49328,I
70  POKE 49330,A + 1
80  POKE 49330,A
90  PRINT X, PEEK (49329)
100  POKE 49330,A + 3
110  NEXT I
120  IF X = 2047 THEN 170
130  READ A
140  POKE 49330,A
150  GOTO 40
160  DATA 16,32,48,64,80,96,112
170  POKE 49331,0
180  END
```

The next BASIC program can program the contents from memory location 10000 to 12047 into the 2716 EPROM.

```
10  REM  PROG. 2716 EPROM
20  POKE 49331,154
30  POKE 49330,9
40  POKE 49331,128
50 X = 9999:A = 0
60  FOR I = 0 TO 255
70 X = X + 1
80  POKE 49328,I
90  POKE 49330,A + 9
100  POKE 49329, PEEK (X)
110  POKE 49330,A + 11
120  FOR J = 1 TO 38: NEXT J
130  POKE 49330,9
140  PRINT X, PEEK (X)
150  NEXT I
160  IF X = 12047 THEN 200
170  READ A
180  GOTO 60
190  DATA 16,32,48,64,80,96,112
200  POKE 49331,0
210  END
```

As A/D converter we have chosen a National Semiconductor ADC0816-type 8 bit successive approximation A/D converter with a 16-channel multiplexer. Figure 4 provides a schematic diagram of the data collection circuit using the MC68705P3 and ADC0816. We have not desided how we will generate the 500kHz clock signal and which serial code we will use, maybe straight winchester. Further we will make a reference circuit to the A/D converter, there only need 5V in power supply as the other chips in the data collection system.

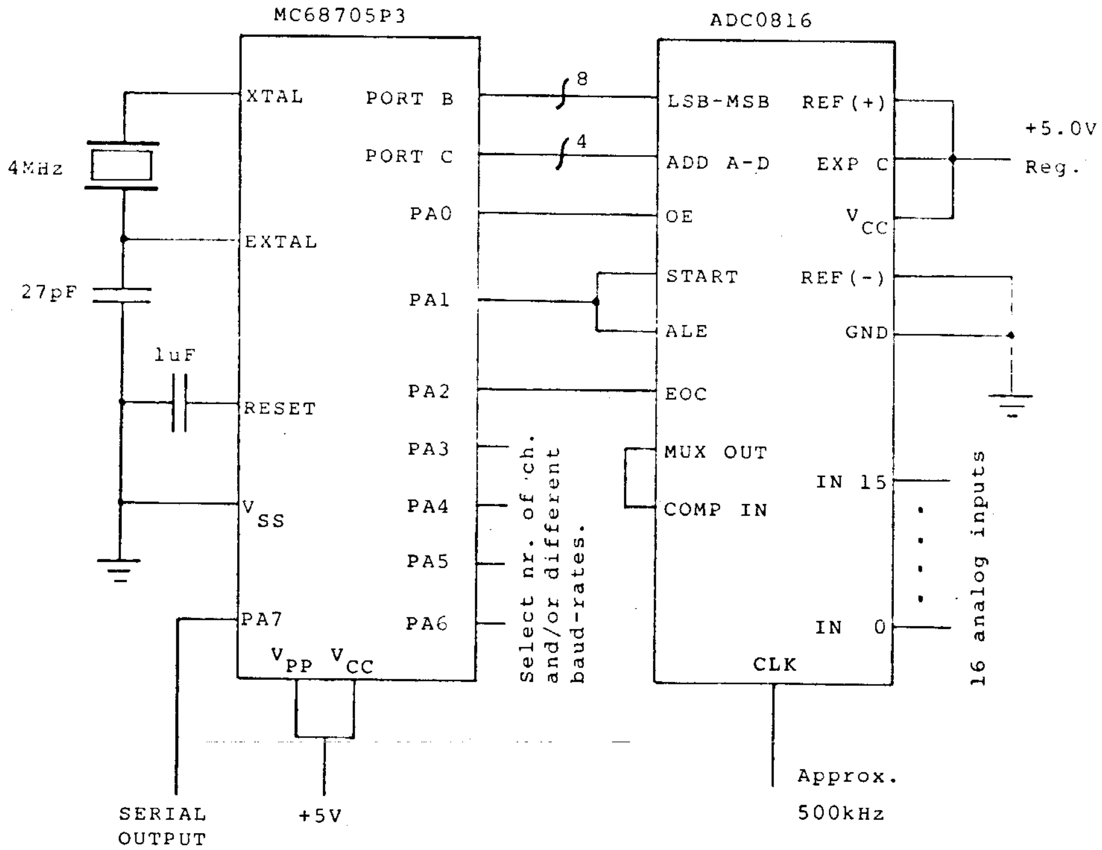


Figure 4. MC68705P3 and ADC0816 interface circuit diagram.

For transmitting digital data over telecommunications links the most commonly-used method is Frequency shift keying (FSK). In order to use FSK, a modulator-demodulator (modem) is needed to translate digital 1's and 0's into their respective frequencies and back again. We will here use the XR2207 FSK modulator and the XR2211 FSK demodulator with carrier-detect capability from EXAR INTEGRATED SYSTEMS INC. The schematic block diagram of the overall system is shown in figure 5.

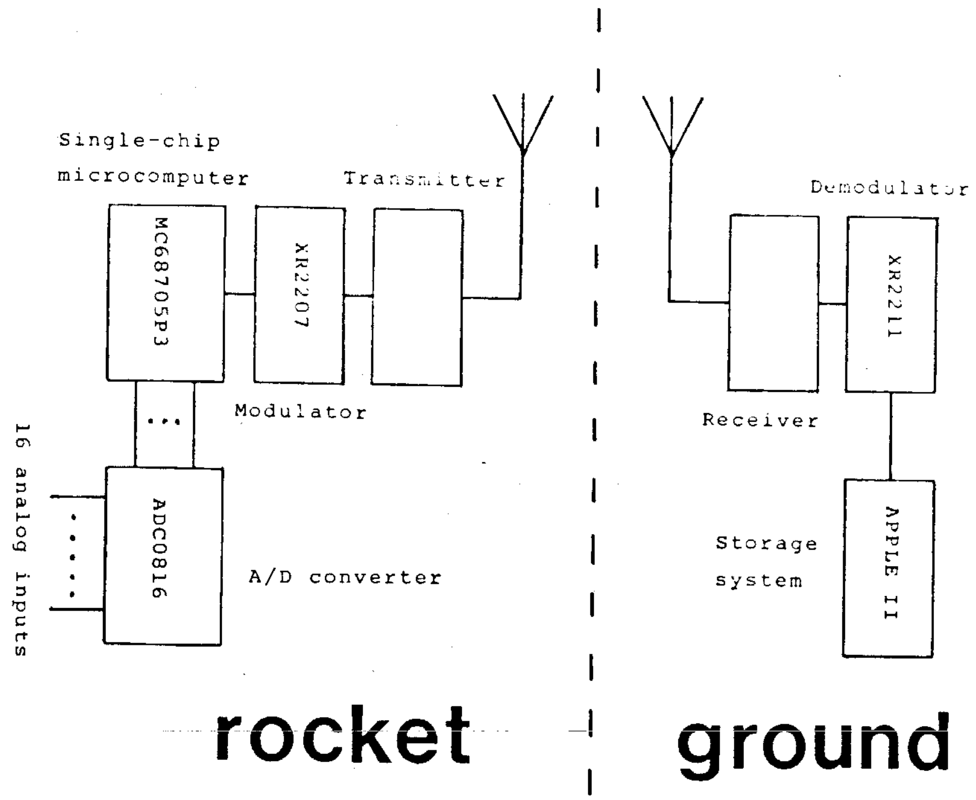


Figure 5. Data collection system block diagram.

for full details, refer to the following data sheets:

MOTOROLA SEMICONDUCTOR: MC68705P3 Advance Information, 1981.

MOTOROLA SEMICONDUCTOR: Design 5ive, Applications, 1982.

INTEL: MCS-48™ Microcomputer User's Manual, 1978-79.

NATIONAL SEMICONDUCTOR: Linear Databook, 1982.

EXAR INTEGRATED SYSTEMS INC.: Application Note AN-01, stable FSK Modems Featuring the XR2206, XR2207 and XR2211, 1981.

EXAR INTEGRATED SYSTEMS INC.: XR2206 Monolithic Function generator, 1977.