A synchro PC interface for the MLS Integrated Approach System

D.Willemsen

Table of contents

Summary

At the moment a lot of research is done to develop a new approach and landing system; the MLS Integrated Approach System (MIAS). This system integrates the Microwave Landing System (MLS) with the Differential Global Positioning System (DGPS). It's purpose is to increase redundancy and subsequently the safty and maybe to replace the expensive Distance Measuring Equipment (DME).

Because all the measurements of MLS and DGPS are obtained with different antenna's, having different places on the airplane, corrections need to be made; for else the aquired accuracy for Cat III landings can't be met. These corrections are made with the use of attitude information, which are available in analogue form. In order to get the attitude information digital, a synchro to pc interface has been developed. Out of three alternatives, the one with the highest accuracy was chosen. This alternative consisted of buying a convertorchip and designing a circuit to make the digital information suitable for pc use.

After writing the supporting software, the interface has been built and tested and proved to be working satisfactory for usage in the MIAS test aeroplane.

Chapter 1. Introduction.

At the moment a lot of research is done with the objective to get better and safer navigation systems for aeroplanes. One of the latest idea's is integrating navigation systems. MIAS is an example of such an integration.

MIAS stands for MLS Integrated Approach System and integrates MLS with Differential GPS. MIAS is designed in order to make a safe and redundant navigation system. When using MIAS as a navigation system it is possible that the DME/P will become obsolete. MIAS has the advantage that it can combine the good qualities of both navigation systems.

When using MLS and GPS together, one needs the attitude angle information, so one can compute a virtual antenna, because the MLS and GPS antenna's aren't on the same place on the aeroplane. This report will describe the design and realisation of a pc card, whichs convert the analogue angle information to it's digital equivalent. This is nescasary because the angle information is needed for futher calculations. Cost and accuracy were the two criteria used for developing the card.

In chapter 2 there will be a short outline of the MIAS system. Chapter 3 and 4 will discuss the conversion and it realisation in hardware. The testing will be discribed in chapter 5 and the 6th chapter will give some errorcalculations and some comments on the test aircraft. This chapter will be followed by conclusions and recommandations in chapter 7.

Chapter 2. MIAS system description

MIAS is a navigation system in which MLS and DGPS data is used in a hybridised way (see figure 2.1.). By using data of both MLS and DGPS, there will be an over-determined system of equations, from which the position of the aircraft can be derived. This overdetermination causes the system to be highly accurate and also useful for integrity checks.

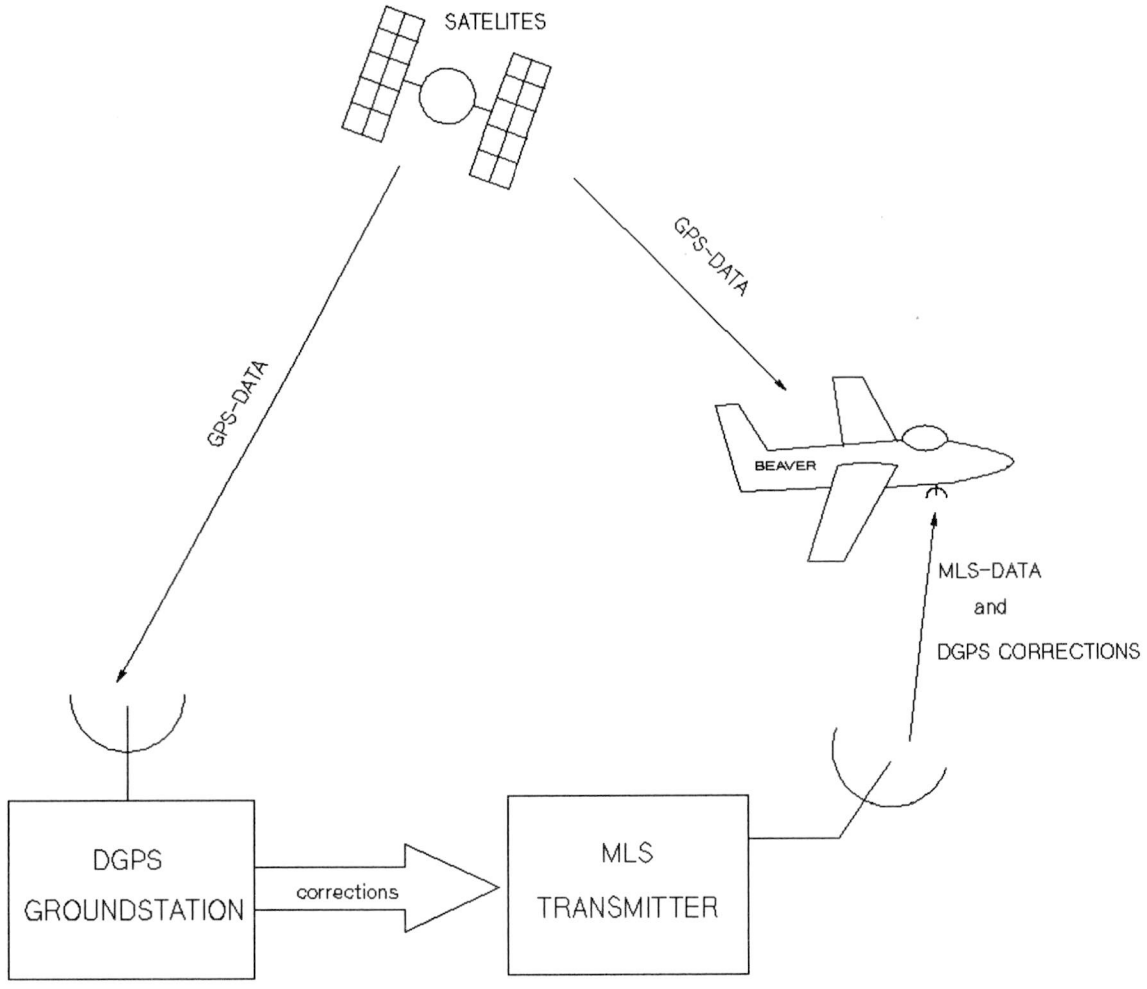


figure 2.1. The MIAS system concept

The main reasons for developing MIAS is that by using MIAS one can get a safer system. A second advantage is that the DME equipment might become obsolete, which means a large

reduction in cost. In the next two paragraphs there will be a short discussion of MIAS system components and a description of the attitude determination.

2.1. MIAS system components.

The two main components of the MIAS system are the Microwave Landing System and the Differential Global Positioning System, which will be discussed below.

2.1.1. The Microwave Landing System (MLS)

MLS is a landing system which calculates the position of the aeroplane by determining the distance r and the angles Θ and Φ. The distance r is the distance between the aeroplane and the DME groundstation and is determined by measuring the transmission time aeroplane-DME groundstation-aeroplane.

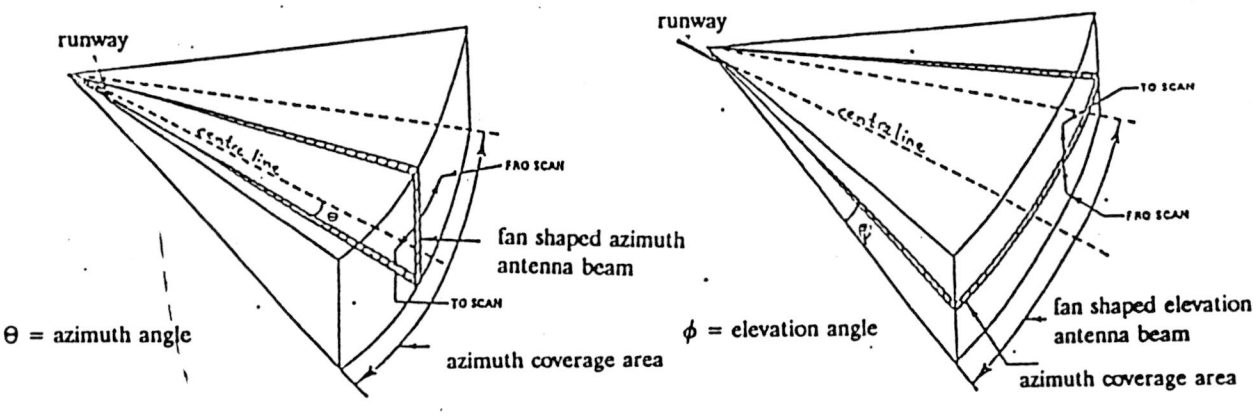


figure 2.2. Fan shaped beams for azimuth (a) and elevation (b) determination

The angles Θ and Φ are called azimuth and elevation respectively. The elevation is obtained by producing a beam which is narrow in the vertical, and wide in the horizontal plane (see figure

2.2). The beam is moved with a constant speed between two borders. At the borders the beam changes direction. The two directions the beam can have are called TO and FRO scans. The borders of the scan area have a fixed value and when the time difference between the TO and FRO scan is available, the elevation can be determined. In the case of the azimuth the same procedure applies, with this difference that the beam is dimensioned in another way (see figure 2.2.).
The azimuth and elevation are transmitted to the aeroplane along with other information like the position of the antenna's, information about the runway, weather specifics, etc. This information is transmitted in two formats; the basic datawords and the ADW's. In MIAS the ADW's will be used to send DGPS correction data to the aeroplane.
More about MLS can be read in references [1] and [2].

2.2. Differential Global Positioning System (DGPS)

GPS is a satellite navigation system. To calculate the position of a user, one needs signals coming from four satellites. Three are used to calculate the actual position and the fourth is used to calculate clockcorrections. GPS suffers from errors like selective availability, ionospherical- and tropospherical delays, multipath, etc. All these sources of errors causes the system to have an accuracy of 10 to 20 meters (95% level) in the case of PPS and 100 meters in the case of SPS. When used in a differential mode, GPS uses a reference station with a known position and at this station the slowly varying errors are estimated and transmitted to the user (see figure 2.3.).

The accuracy can be 2 to 8 meters when the user is near the reference station. The accuracy is reduced as the user-reference station separation increases (spatial decorrelation) and as time passes (temporal decorrelation). An additional benefit is that the DGPS reference station acts as a monitor and can inform the user of problems that might arise in the use of the satellite signals.
More about (D)GPS can be found in references [3] and [4].

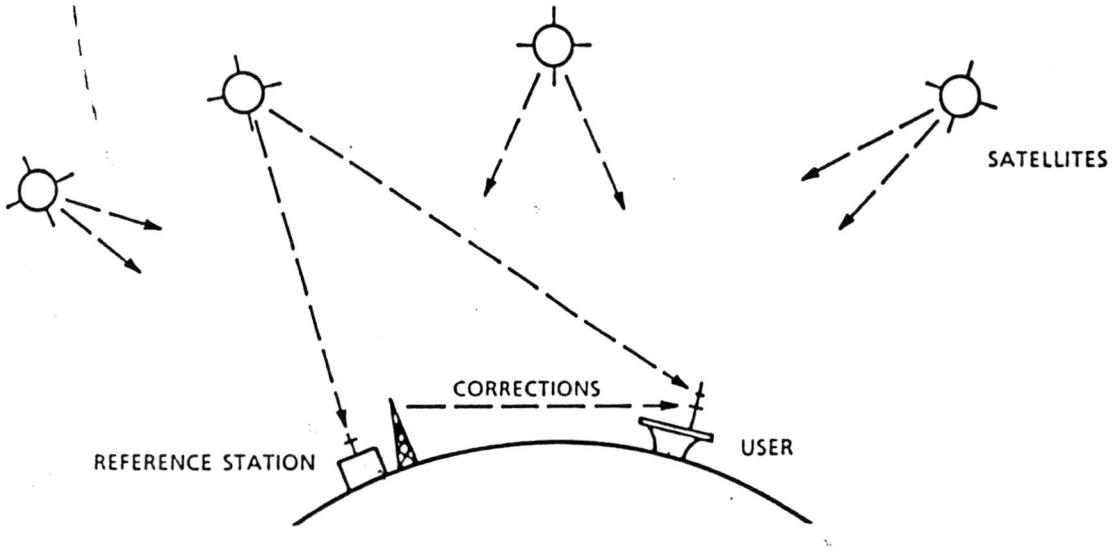


figure 2.3. Differential operation of GPS.

2.3. Attitude determination for MIAS

The accuracy of the position of the aircraft is to be determined with an accuracy of decimeters for Cat III landings. Calculating the position of the aircraft with the measurements of MLS, DGPS and DME will result in an undefined point on the aircraft somewhere between the MLS, DGPS and DME antenna's. This is because all three systems use different antenna's. The MLS, DGPS and DME measurements should be adjusted to take their different positions into account. In order to make these adjustments one needs the attitude information. With this information one can calculate the position of a virtual antenna which can be placed anywhere on the aircraft [9].

The attitude information consists of three angles: roll (ξ), pitch (ϕ) and heading (β) (fig.2.4) Each of these angles comes from a synchro, which is device that transforms the angle information obtained by the gyroscope into a electrical signal. See also appendix A. In the test aircraft that was used for testing MIAS, these angles where only available in analogue form and MIAS needs

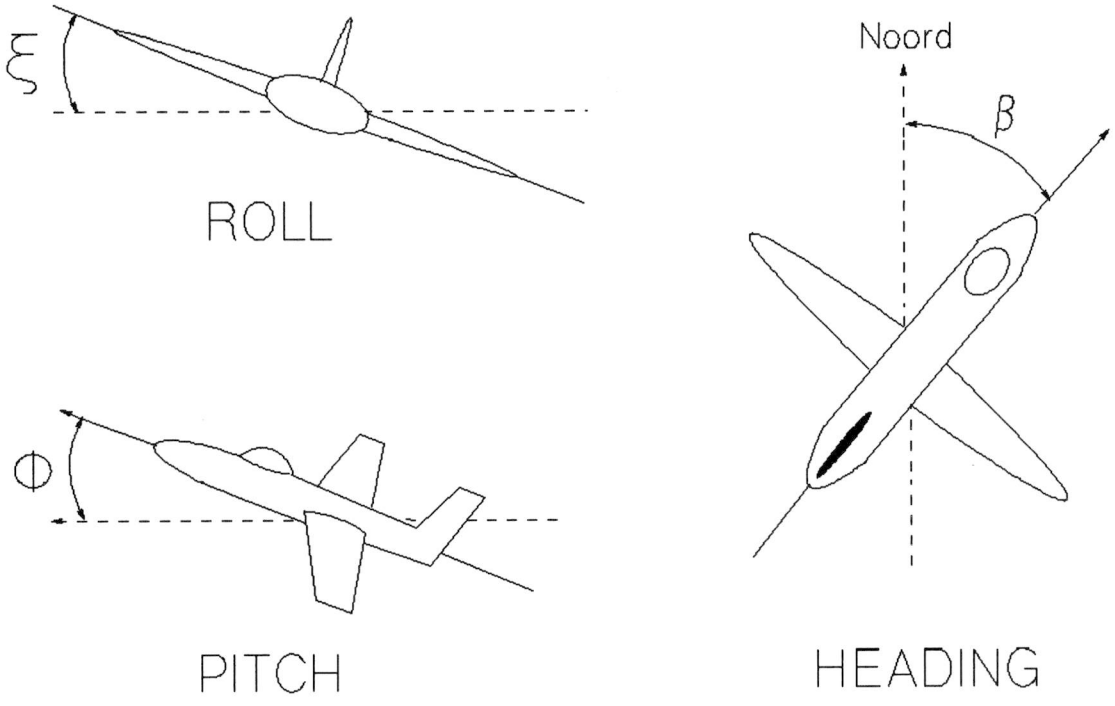


figure 2.4. Roll, pitch and heading

them digital. The conversion from analogue to digital as well as the implementation for PC use forms the basis for this report and will be discussed in the next chapters.

Chapter 3. Conversion of the attitude information.

As could be read in chapter 2, we need to transform the analogue information to it's digital equivalent. The analogue information comes from synchro's. A synchro is a device in which a rotor coil, with a AC voltage of fixed frequentie and amplitude, is connected to a gyroscope. Three stator coils are positioned around the centre of the synchro at equal angle's of 120°. By rotating the rotor coil, voltages are induced in the statorcoils. These voltages are a measure for the angle (see also appendix A).

There are three ways to realize the total card:

* buying a complete card with supporting software
* use an A/D convertor and design a sampling circuit which samples the analogue signals so that they are suitable for A/D conversion
* use a conversionchip and design a circuit to make the digitalized signals suitable for pc use

When applying the two criteria, cost and accuracy, mentioned in chapter one, the first two alternatives can be rejected. The first one because it is too expensive. Alternative two and three cost about the same, but option two isn't very accurate compared to option three. This is because the different parts for conversion aren't integrated as is the case in option three. Another important reason is that option three is available on the market and thus is extensively tested. So option three has been chosen to be implemented.

Paragraph 3.1. will explain the principle used by conversion chips and the second paragraph will give a description of the chip chosen for implementation

3.1. The analogue to digital conversion.

The main principle of this conversion is shown below (fig.3.1.). As we can see, the three signals coming from the three stator coils (S1, S2, S3) are rearranged, so that only the angle Θ remains. Doing this, we obtain the signals:

sinθ and cosθ **(5.1)**

This is what is done in the first block

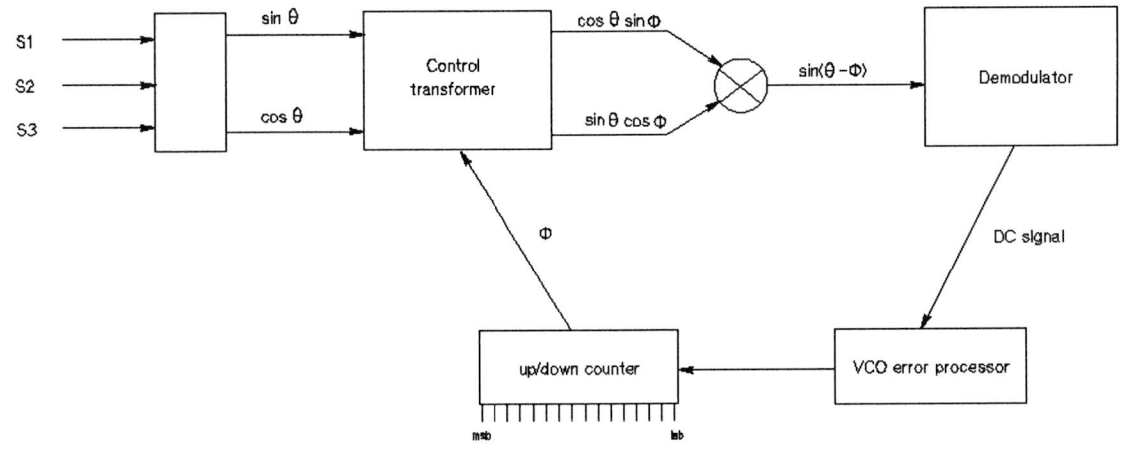


figure 3.1. The principle of synchro conversion

In this discussion the cos ωt carrier term will be ignored, because it will be removed in the demodulator. Next the sin Θ and cos Θ will be multiplied with cos ϕ and sin ϕ which yields:

sinθcosϕ **(5.2)**

cosθsinϕ **(5.3)**

This is done by the control transformer. These outputs are fed to a subtracter, so that the input fed to the demodulator is:

sinθcosϕ−cosθsinϕ=sin(θ−ϕ) **(5.4)**

The righthand side of this expression is a signal with an amplitude which is proportional to the difference between θ (the angle to be digitized) and ϕ (the angle stored in digital form in the up-down counter). The output of the demodulator is an analogue DC level proportional to (5.4), which is the sinus of the error between θ and ϕ. This signal serves as a input-signal for the VCO and this VCO produces clock pulses that are counted by the up/down counter. The sense of the error (θ too high or θ too low) is determined by the polarity of (θ - ϕ) and is used to generate a counter control signal, which determines wether the counter increments or decrements.

3.2. The RDC 19220 synchro convertor.

The chip used for implementation in this project is the RDC 19220 chip from DCC. This chip contains all the elements described above. The RDC 19220 offers some very useful options [5]:

* the data generated by the chip is always fresh, i.e. it is continuously updated
* it is possible to "freeze" the data, so it can be read and used for further processing
* it is low cost, each chip costs about fl.300,-
* the angle is represented by 16 bits, which means we can reach an accuracy of $360/2^{16}$

The chip wasn't originally designed for synchro use, but with the use of a Scott-T-transformer (fig.3.2.) it is very easy to convert the synchro signals. The Scott-T-transformer is nothing more than a resistor network, which converts the three signals from a synchro to four signal which are suited for the chip. This circuit is nothing more than a voltage deviding network. In this figure -S, S, sin, -C, C and cos are the input signals of the chip [5] and S1, S2 and S3 are the synchro signals.

In appendix B the settings of the chip are outlined and all the resistor values will be given.

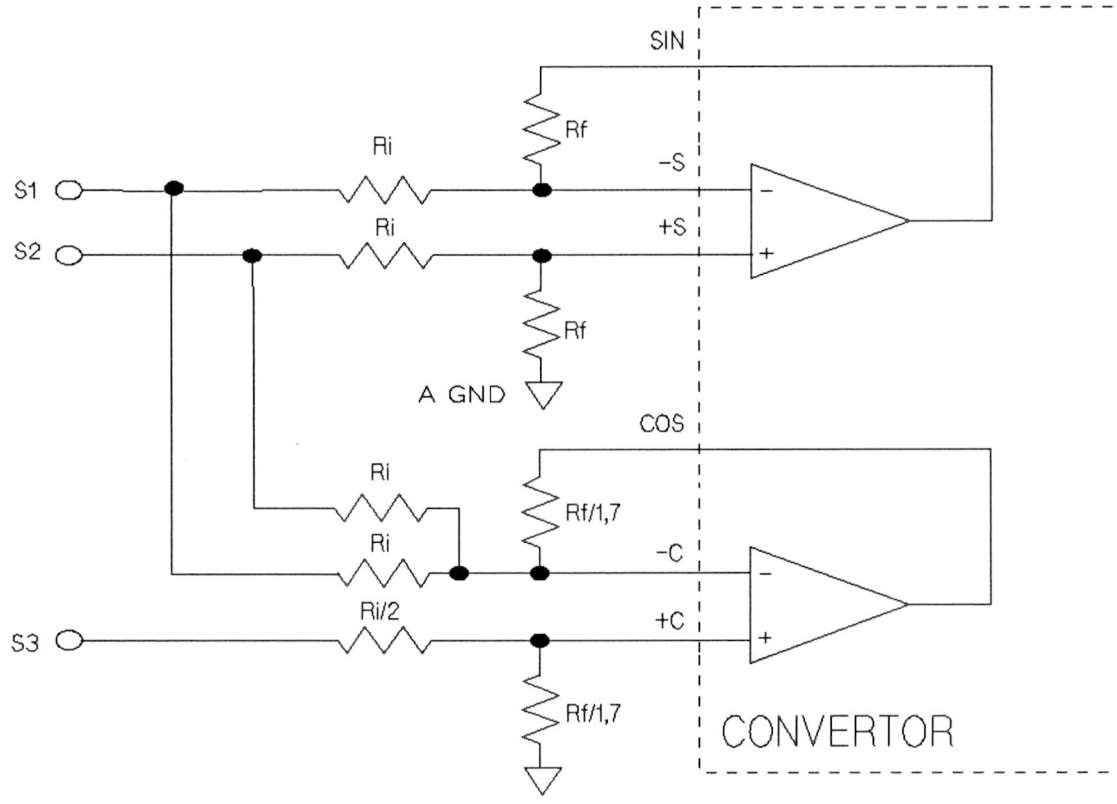


figure 5.2. The Scott-T-transformer

Chapter 4. The control electronics for the synchro to PC interface.

In the previous chapter we saw how we could transform the analogue synchro signal to it digital equivalent. In order to make this digital signal available for a computer, one needs to design a card which does just this. When we look at the I/O channel connector of the IBM computer (fig. 4.1.) one can see all the pins and their functions. The following signals will be used for making the control electronics:

* SD0 - SD7 : the 8 bit data signal
* I/O ready : tells the processor when the process is ready
* I/O read : signal generated by the processor to stress that there has been a read instruction
* SA0 - SA9 : the addressing of the card
* (B)ALE : indicates if a valid address has been used in the instuction

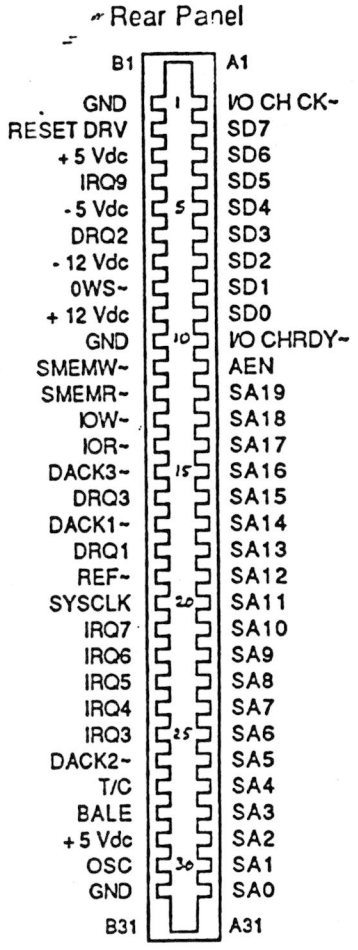


figure 4.1. The I/O channel connector

There are four distinctive sections on the card, each having their own function:

* addressing section
* timing section
* selection section
* output section

These four sections will be discussed in the next four paragraphs and the actual components and their settings are given in appendix C.

4.1. The addressing section

In order for the computer's processor to be able to access a card, an address has to be assigned to the card. In the IBM specifications [6] the address $300 through $30F are assigned to prototype cards. The address of the card can be set by using jumpers. At each chip one can freeze the lsb and the msb seperately, so in order to select the msb's and lsb's of three chips, one needs six combinations; this can be achieved with three adresslines and a multiplexer. Another adress line is needed to freeze the data, else it would alter when reading it into the computer. The binary representation of $300 is:

|0011|0000|0000|

and that of $30F is:

|0011|0000|1111|

so the first eight bits have to set by the jumpers, but because we are only using ten address lines, (this is because a $3 can be set by only using two jumpers and we're not using adresses bigger than $30F), the first two are omitted and the jumper settings will be:

| 11|0000|

with each digit representing a jumper. The card is now accessible by using $300 through $30F. When the processor sends a address to the card, the card has to determine if it is a valid address (this is because the parallel port is used, so all the different addresses for the different cards are send to this port and the card has to determine when it is addressed). This is done by using a comparator which compares the incoming address with the one set by the jumpers. It will also check on the (B)ALE signal and when the address is correct and the (B)ALE signal has a falling edge, then it will give an okay signal.

4.2. The timing section.

The timing section is the section which interconnects the other three sections. This means that it will check the various component and generate the go signals. When the I/O read signal is low and the comparator gives an okay signal, then a pulse will be given to the other sections. (in a way this is the internal clock of the card looks like the I/O read signal given by the PC).

In the case when there is a pulse, the I/O ready pin will be held low for 388ns. This is done because the synchro convertor's data is not valid until 350 ns after the INHIBIT signal freezing the chips data is invoked. When I/O ready returns to the high level, the data on the bus reads on the next negative clock edge. (fig. 4.2.) This 388ns delay is realized by using a mono stable multivibrator. (388 ns is the value which is set by the multivibrator, because it wasn't possible to realize a time of 350 ns, in this case you can be certain that the data is valid)

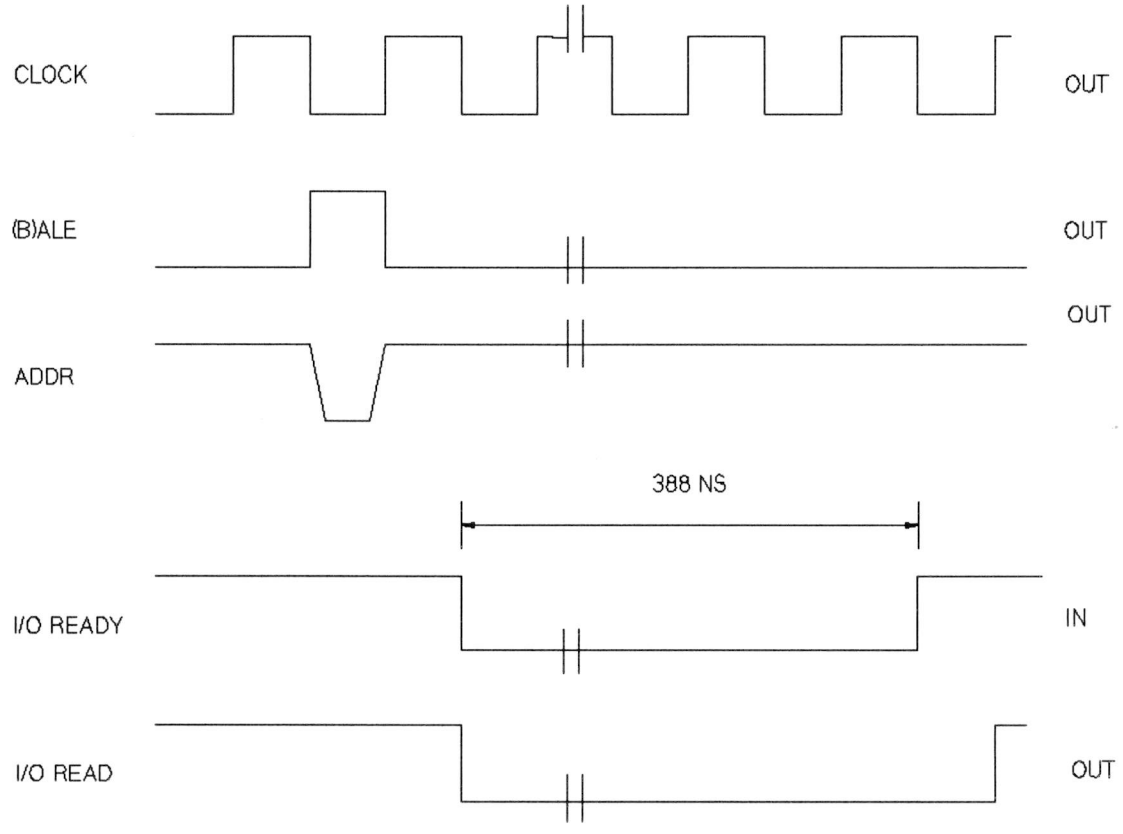


figure 4.2. The signals and their timing

The values of the three variable address lines and the signal going to the I/O read pin are stored in a 3 state buffer and they are passed through if the internal clock give a go signal.

4.3. The selection section.

The selection of the three chips is done by using a multiplexer. We have three address lines (A1 - A3) to select the different chips and each chip has two values, i.e. the most significant byte and the least significant byte. With three lines there are eight possible combinations, but we only need six. The A0 line is used to set and reset the chips. Table 4.1. gives the addresses and the bytes which are selected.

A3	A2	A1	selected chip
0	0	0	1 (bit 1-8)
1	0	0	1 (bit 9-16)
0	1	0	2 (bit 1-8)
1	1	0	2 (bit 9-16)
0	0	1	3 (bit 1-8)
1	0	1	3 (bit 9-16)
1	1	1	none
1	1	1	none

Table 4.1. Selected bytes

4.4. The output section.

The card has it's own databus, which consists of eight datalines. When a byte is selected, it will be put on the bus and by using a octal buffer with a 3-state output this byte is put on the

computers bus when the I/O ready signal turns high and the clockpulse turns low. (see fig.4.2). In order to read an angle, we need the two different bytes of one chip. These two bytes will be combined and a angle calculation will be made (see chapter 5).

When we link the four sections together, we will get the following figure (see fig.4.3).

Appendix C will give all the components and there settings as well as the realisation on a printboard. The total costs of making the printboard and buying the components is approximately fl.1100,-.

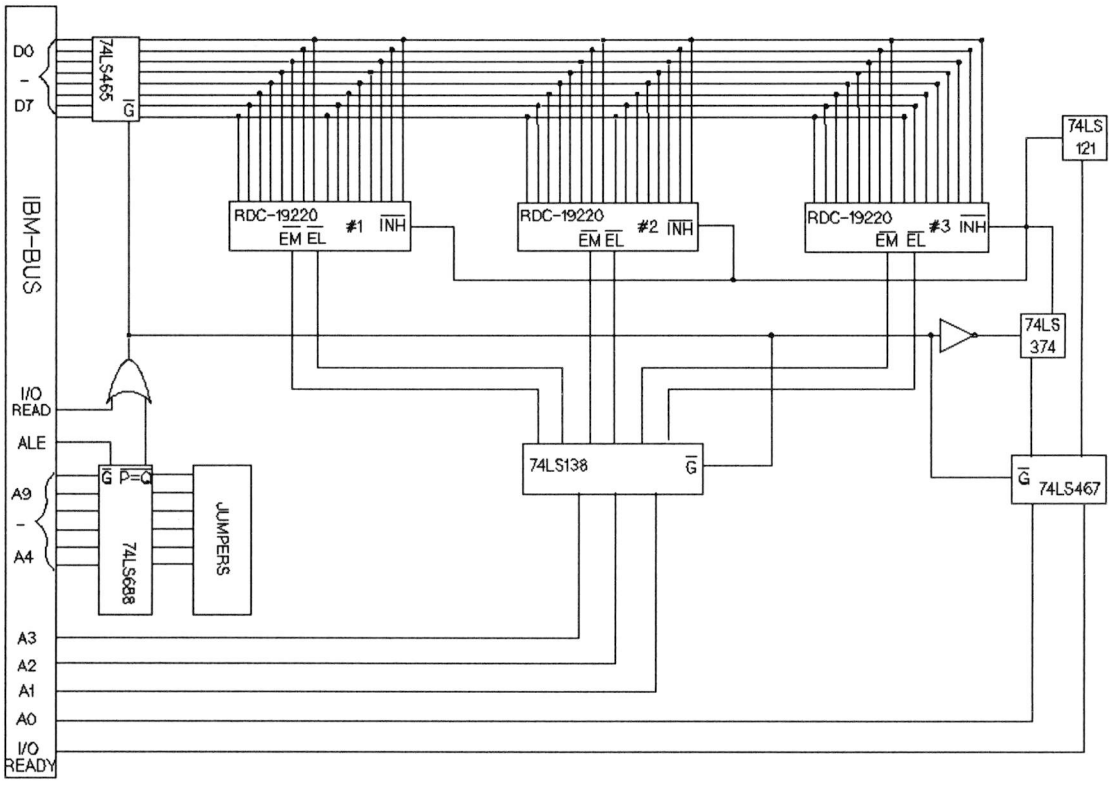


figure 4.3. The pc card

Chapter 5. Software and testing of the prototype card.

In the previous chapters we have designed a card which can convert analogue synchro signals into it's digital equivalent. Before we can use the card in the aeroplane, appropriate software has to be written and the card must be tested. This is done in the next two paragraphs.

5.1. The cards software.

The function of the software is to calculate an angle from the two bytes which are obtained from the card. This angle can be used for futher calculations [7]. The first problem is to link the two bytes together, so we can calculate the amount of steps. This is done with the next equation:

$$\mathbf{lsb + msb * 256} \tag{5.1}$$

We have sixteen bits to represent 360 degrees, so the equation for the stepsize becomes:

$$\mathbf{stepsize} = \frac{360}{2^{16} - 1} \tag{5.2}$$

You have to devide 360 by 2^{16}-1, because if you devide only by 2^{16}, both 0 and 360 degrees can occure. When we multiply (5.1) with (5.2), we will get the actual angle. Off course this is an uncalibrated angle. To calibrate the card one has to calculate the number of steps for a known angle and use this as a bias for the other calculations. For exemple, when the angle is zero degrees, the amount of steps also has to be zero. So when in this case we measure a amount of step of 26000, it is obvious that the bias is 26000.

It is also nescesary to do some error testing. This is done in three ways:

* each byte will be read twice and if they differ too much, the measurement has to be done again. To be more specific, the two most significant bytes must be the same and the least significant bytes can only have a different least significant bit.
* the amount of steps has to be checked if it lies in between 0 and 2^16.
* the angle has to be in the next interval:
  $0 \leq \text{angle} < 360$

Taking al those requirements into account, we come to the next structure diagram (fig. 5.1.). In appendix D the realisation in TURBO PASCAL is given and discussed.

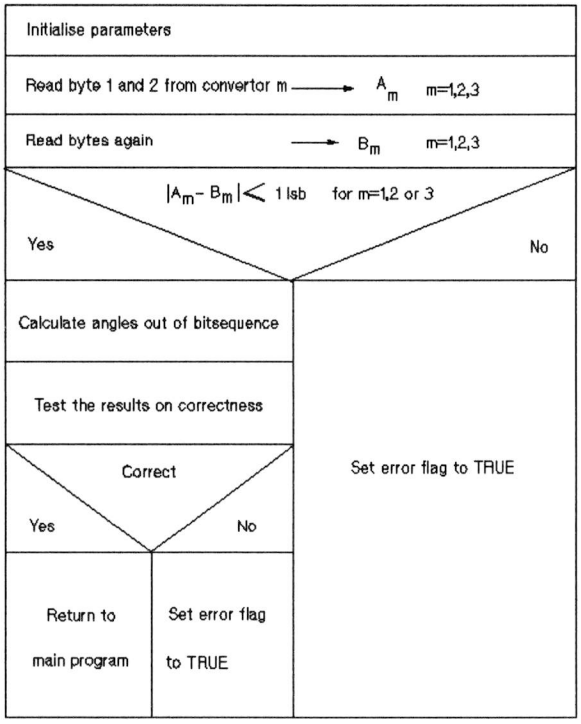


figure 5.1. Structure diagram

5.2. Testing of the card.

The testing of the card wasn't to difficult a task. First a testing program was written (see appendix D). Second the card, without convertor chips, was inserted in the computer and the following signal were measured with a logic analyser:

* INHIBIT signal
* the comparators okay signal
* the adress lines
* the go signal
* the chips enable signals

The results obtained were as expected (like fig.4.2) and the control electronics were working okay. Unfortunately the logic analyser didn't have the possiblity to make printouts, so none will be given here.

Knowing that the control electronics worked well, the convertor chips were implemented and tested. This testing was done by using a synchro which was connected to the card. The synchro could be manualy adjusted. With the software the card was read and the angle was printed on the screen. There were a few things which were looked at:

* to see if after setting the bias, the scale was linear (this was done by looking if the calculated angles corresponded with the angles set on the synchro)
* the fluctuation of the angles

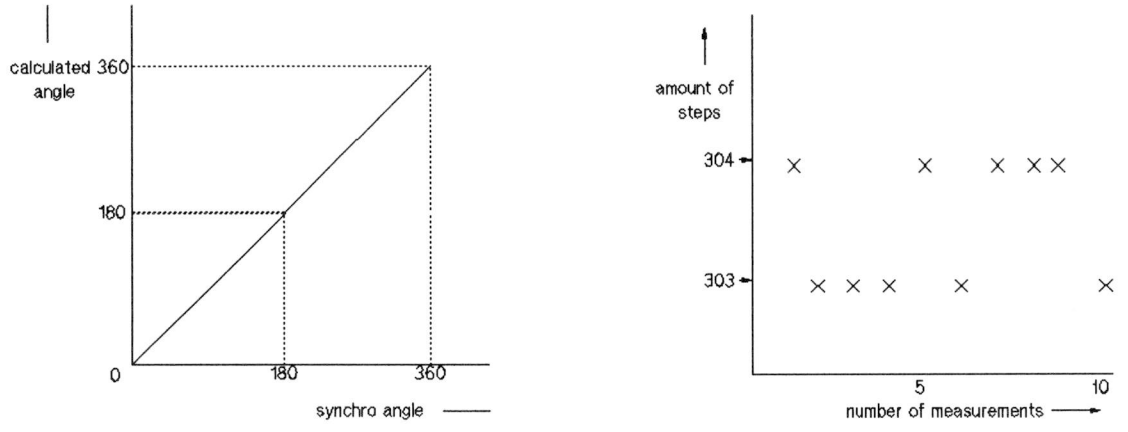


figure 5.2. Testresults

The results were that the scale was very linear; a synchro angle of 84 degrees also had a calculated angle of 84 degrees. The fluctuation when measuring the same angle a few times was one stepsize. This was expected because off the accuracy of the convertor chip [5]. See also figure 5.2.

Chapter 6. Implementation in the test aeroplane.

After designing and testing the card it has to be put into the testing aeroplane. This chapter gives a short outline of the signals which are availible in the aircraft and also a rough estimation of the error made.

6.1. The test-aeroplane.

The aeroplane which is used for testing the MIAS concept is the Beaver. This is the aircraft from the faculty of aereonautics. The roll, pitch and heading information is generated by gyroscopes, the synchros connected to these gyroscopes generate analogue signal which represent the angle measured by the gyroscopes. Normally it isn't possible to get hold of these signals because the signals go directly from the sensors to the instruments.

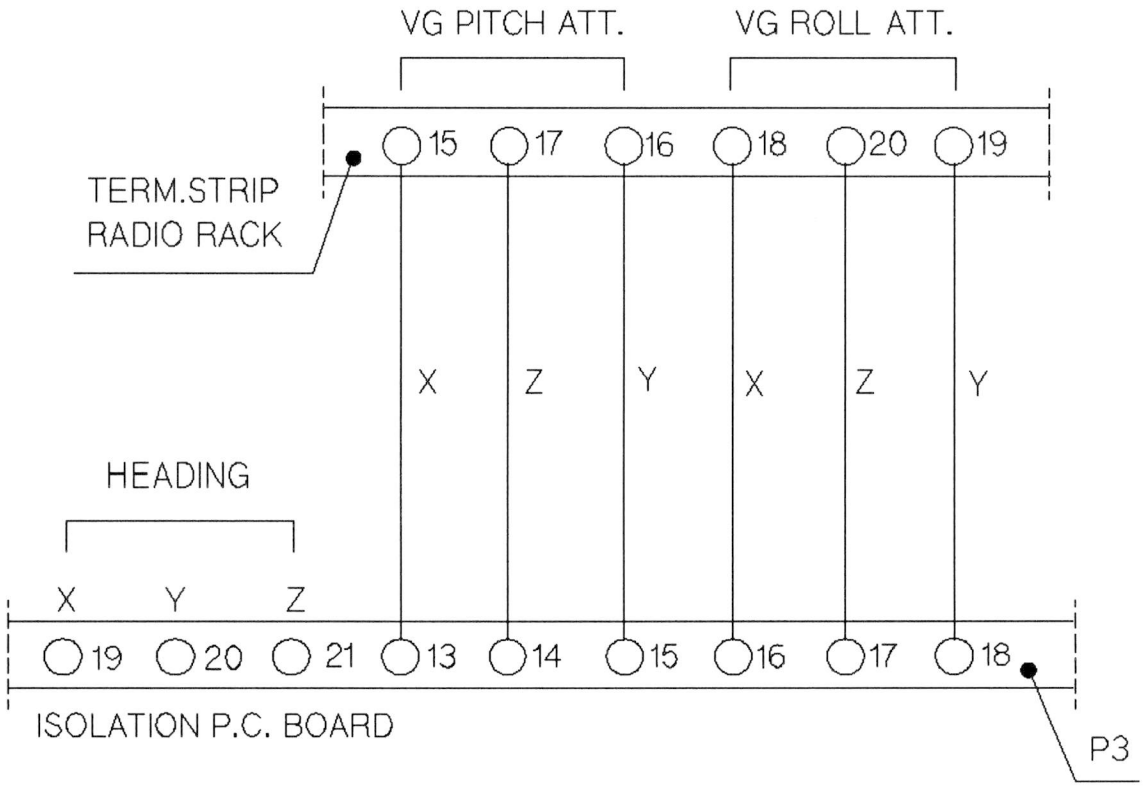


figure 6.1. The connection board in the Beaver

In the Beaver there have been made some provisions, so that these signals can be used for

experiments. In this case it means that all the synchro's are double, one is used for the instruments and the other can be used for experimental setups (if this wasn't the case the card could still be used by making a parallel connection, because of the high input impedance of the card, the instruments wouldn't see the difference). The different synchro signals are fed from the internal radio rack to a external pc board, where they can be used for testpurposes. (fig.6.1)

When connecting the signals to the card you don't need to worry about ground loops, because the gyro's aren't connected to ground. The power to support the computer can be supplied by the aircrafts internal power system.

6.2. Error calculation.

There are three places errors can occur; in the gyro's, in the synchro's and in the convertors (from the card). When an aeroplane is in it's landingsfase (fig.6.2.) the error in the pitch angle is the most important error, so the error is estimated with (6.1), in which $\Delta\delta$ is the error in the pitch angle and ΔS is the error in the height. A is the distance between the wheels and the nose of the aeroplane. When we take A to be 20 meter (this is for a large plane) we find, when using (6.1), that a change of 0.1 degrees of the pitch results in a 3.5 cm deviation of the heigth

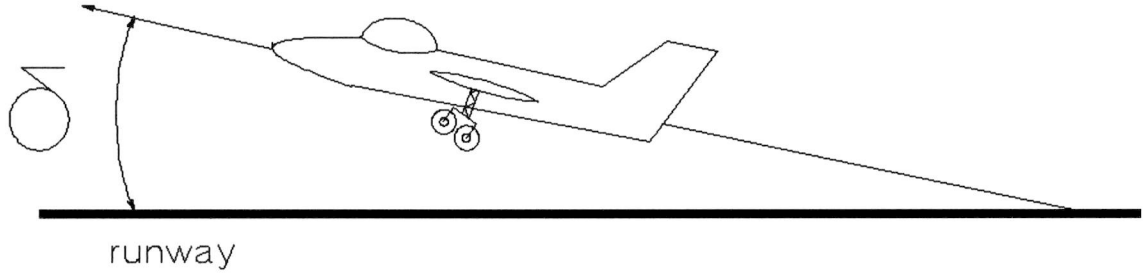


figure 6.2. The aeroplane in it's landingsapproach

$$\sin(\Delta\delta) = \frac{\Delta S}{A} \qquad (6.1)$$

There isn't a fixed value for the error made in the gyroscopes, this is because the gyro is very sensitive to gravity. The synchro has a standard deviation of 7 min and this is 0.12 degrees. The

standard deviation of the synchro-convertor is 8 min + 1 LSB [5]. When using 16 bits the standard deviation becomes 0.138 degrees. The standard deviation of the instruments is about 0.4 degrees (this figure is a value obtained from the aeroplane's experts). So the total standard deviation is 0.4 + 0.138 and this would lead to an error in the height of approximately 20 cm. MIAS was designed with the objective of having an error of 60 cm. More about aeroplane instrumentation can be read in [8].

Chapter 7. Conclusions.

When integrating two different systems to get a better accuracy, there have to be made some corrections concerning the position of the antenna's of the induvidual systems; there has to be one virtual antenna which uses measurements coming from both systems. In order to make these corrections one needs the attitude information, which only is available in analogue form. To be able to process this information digital, a synchro to PC interface is nescessary.

With the alternative given in this report, it is relatively easy to built a synchro pc interface for MIAS. With less than fl.2000,- it is possible to make an interface which has a good accuracy. The results of testing were that the interface gives the real angle with an standard deviation of only 0.538 degrees. The accuracy is about 20 cm for an aircraft of about 20 m. The calculations of the accuracy are rough theoretical estimates, so further research could be done to know exactly how big the fault is.

The card has been built and worked satisfactory and can be used for implementation in the test aeroplane.

List of References

[1] Kelly, Robert J. and Redlien, Henry W., "Microwave Landing System: The New International Standard", Advances in electronics and electron physics. Vol 57

[2] Willigen van, D. and Vlietstra, E.P.M., "MIAS, The integration of MLS with DGPS/DLORAN-C", Delft University of Technology, January 27-29, 1992

[3] Vlietstra, E.P.M., Thesis report:"MLS Integrated Approach Systems - MIAS", Delft, June 1991

[4] Enge, Per K., Kalafus, Rudolph M. and Ruane, Michael F.,"Differential operation of the Global Positioning System", IEEE Communications Magazine, July 1988 - Vol.26, No.7

[5] RDC 19220 chip specifications, DDC. ILC data device corporation, Data convertors product catalog, 1991

[6] Markowsky, G.,"A comprehensive guide to the IBM Personal Computer", Prentice-Hall, Englewood Cliffs, 1984

[7] Meijer, M., Thesis report, "Development and testing of the airborne part of the MLS Integrated Approach System.", Delft, May 1993

[8] Abbink, F.J., "Vliegtuiginstrumentatie I", Delft University of Technology

[9] van Leeuwen, René G.A., Thesis report, "Geometry aspects of the MIAS-project", Delft, November 1992

Appendix A. The working of a synchro.

A synchro is a device which transfers angles. The advantages of synchro's are that they can work under extreme surrounding conditions like pressure, humidity, vibrations, etc. They can also rotate through "the zero" , which means that they can make more than one rotation without complications. They also have a good accuracy. Because of these qualities, they are used in aeroplanes a lot.

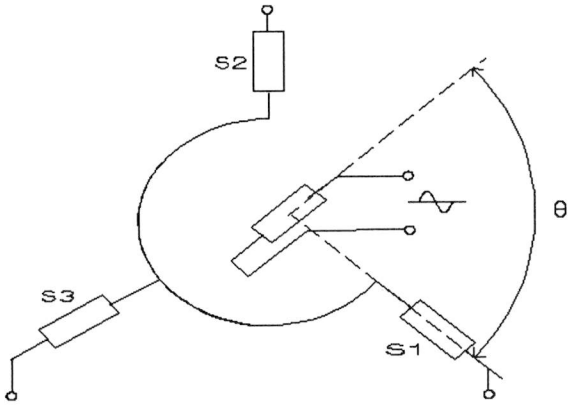


figure A.1. The synchro

An A.C.-synchro consists of a rotor coil with an A.C. voltage with fixed amplitude and frequency. The rotor coil is connected to a gyroscope. Placed around the rotor coil are three stator coils each spaced 120 degrees (see fig. A1). In the three stator coils (S1, S2, S3) a voltage will be induced. When the rotor coil has a voltage of $V*\sin(\omega*t)$, which in aeroplanes is usually 26 Volts RMS, then a voltage will be induced which is dependent of the angle between the rotor and the stator coils (see fig. A1). If the angle θ is the angle which the rotor coil makes with stator coil 1, then the induced voltages in S1, S2 and S3 are given by:

$$V_1 = V * \cos(\theta) * \sin(\omega * t) \quad \textbf{(A.1)}$$

$$V_2 = V * \cos(\theta + 120°) * \sin(\omega * t) \quad \textbf{(A.2)}$$

$$V_3 = V * \cos(\theta + 240°) * \sin(\omega * t) \quad \textbf{(A.3)}$$

Appendix B. The settings of the convertor chip.

This appendix will discuss the settings of the convertor chip. This is done with the aid of the chips specifications [5].

The Scott-T-transformer

The input voltages of the chip are 2 V_{rms} so the input signals have to be converted from 26 V_{rms} (the voltage coming from the synchro) to 2 V_{rms}. This is done by a Scott-T-transformer. The 26 V_{rms} on the nodes S1, S2 and S3 (see fig.3.2) is converted to 2 V_{rms} on the pins -C, +C, cos, -S, +S and sin. The following equation applies to determine R_i and R_f (B.1) [5]:

$$\frac{R_i}{R_f} * 2V_{rms} = 11{,}8V_{rms} \qquad \textbf{(B.1)}$$

The values which have been chosen are:

$R_i = 330k\Omega$
$R_f = 56k\Omega$

These values have to be very accurate because otherwise the transformation will go wrong.

REF inputs

A synchro also has Reference outputs and these outputs are connected to the REF input pins of the chip. Because the maximum input voltage can only be 4 V_t, the voltage coming from the synchro has to be modified. Figure B.1 shows the circuit which is used for modification and connecton. Node A corresponds to the synchro reference signal of 26 V_{rms} and node REF corresponds to the chips input signal of 4 V_t. When looking at fig.B.1 and bearing in mind that

26 V_{rms} equals $26\sqrt{2}$ V_t, it can be seen that (B.2):

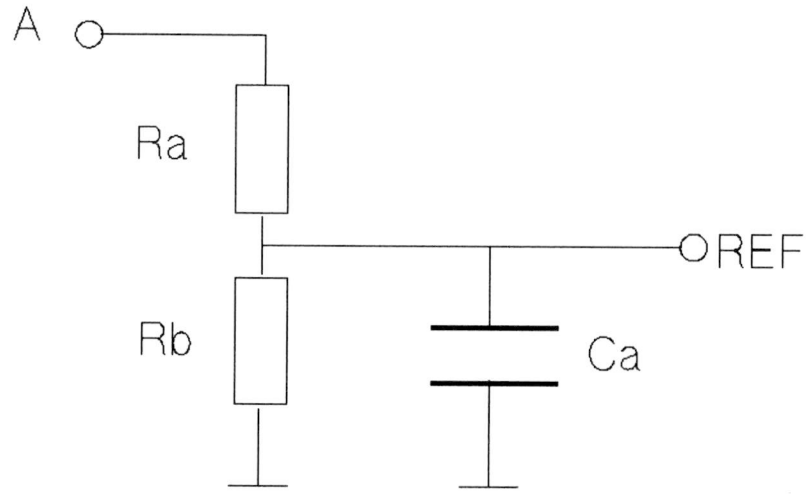


figure B.1. Input circuit for reference signals.

$$\frac{R_b}{R_a+R_b} * 26\sqrt{2} \leq 4 \qquad \textbf{(B.2)}$$

,which after some calculations leads to (B.3):

$$\frac{R_a}{R_b} \geq 8{,}19 \qquad \textbf{(B.3)}$$

The values of R_a and R_b have been chosen to be:

R_a = 82 kΩ
R_b = 8k2 Ω

C_a is used to protect the inputs against high frequency distortions and together with R_a and R_b forms a LPF. With the use of some elementary calculations, it can be seen that the value of C_a has to be smaller then 1 μF; this when the filter has a bandwidth of 400 Hz. C_a has been given

the following value:

$$C_a = 100 \text{ nF}$$

Dynamics of the RDC 19220

The dynamics of the chip determine the chips achievements and can be adjusted by using the following three equations [5]:

$$R_v = 55k * \frac{max.rate}{userrate} \quad \textbf{(B.4)}$$

$$C_{bw} = \frac{3.2 * R_s * 10^8}{R_v * f_{bw}^2} \quad \textbf{(B.5)}$$

$$R_b = \frac{0.9}{C_{bw} * f_{bw}} \quad \textbf{(B.6)}$$

These equations are directly taken from the specifications. When taking R_s=67kΩ, f_{bw}=100 Hz, userrate=21 and max.rate=18 [5] then the following values are found:

C_{bw} = 4n7 F
R_v = 47 kΩ
R_b = 180 kΩ

Voltage peak protection

In order to protect the input pins against peaks in the voltage, diode's have been connected.

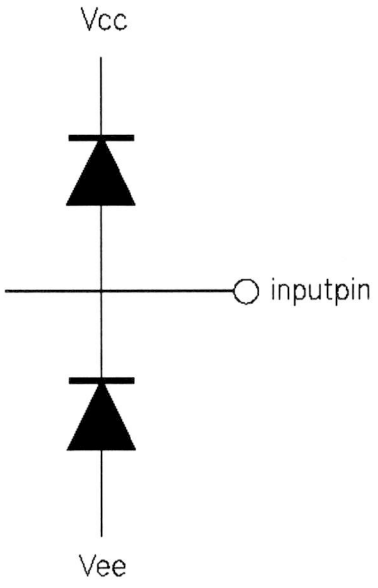


figure B.2. Voltage peak protection

Whenever the voltage gets higher than five volts (absolute value) then the diode's will conduct. These diode's don't influence the working of the chip.

Appendix C. Hardware realisation of the card.

In this appendix all the components and their function and settings will be discussed. There is also an indication given of the section the component belongs to. Figure C.1. gives the complete lay-out of the card with the synchro convertor chips. Figure C.2. gives a listing of all the components and their function tables.

The 74LS688 ic (addressing section)

This is a 8-bit magnitude comparator, which compares the address send with the address set by the jumpers. The (B)ALE signal is connected at the G-pin and when this signal is low and the addresses match, then the P=Q pin will be low. See figure C.2.

The jumpers (addressing section)

When a jumper is inserted, the voltage will be zero volts (logic "0"). When there is no jumper, the voltage is pulled up to five volts (logic "1") by a pull-up resistance. See figure C.2.

The 74LS465 ic (output section)

This ic is a octal buffer with a three state output. So when the G1 en G2 pin are low, then it will pass the from the A-pins to the Y-pins. In all the other cases, the databyte will not be given through. The G signal is generated by the timing section. See figure C.2.

The 74LS138 ic (selection section)

The 74LS138 is a multiplexer. Depending on the A, B, C and G signals it will make one of the outputs Y0 through Y7 low. A, B and C correspond to the three variable address lines. The G2 signal again is generated by the timing section. See figure C.1. and C.2.

The 74LS00 ic (timing section)

This is an ic with a 8 NAND ports. When the P=Q signal from the comparator is high and the (B)ALE and I/O read signals are low, then there will be a low timing signal (the G-signal). Only one ic (the 74LS74) needs a high G signal and this is accomplished by using a NAND port. See fig. C.1.

The 74LS467 ic (timing section)

This ic is similar to the 74LS465 and it makes sure that the signals are only given through when the address is correct.

The 74LS121 ic (timing section)

This is a monostable multivibrator with Schmitt-trigger inputs. It's function is to keep the I/O ready signal low for at least 350ns, this is because, after given the INH signal, the data needs 350ns to stabilize. As soon as the INH signal turns from high to low, the 74ls121 will generate a low signal for 350ns. The time delay can be calculated by using the next formula:

$$\mathbf{t_w(out) = C_{ext} * R_T * ln2} \qquad \textbf{(C.1)}$$

The following values have been chosen:

R_T = 5k6
C_{ext} = 100pF

Which lead to a delaytime of

$t_w(out)$ = 388 ns

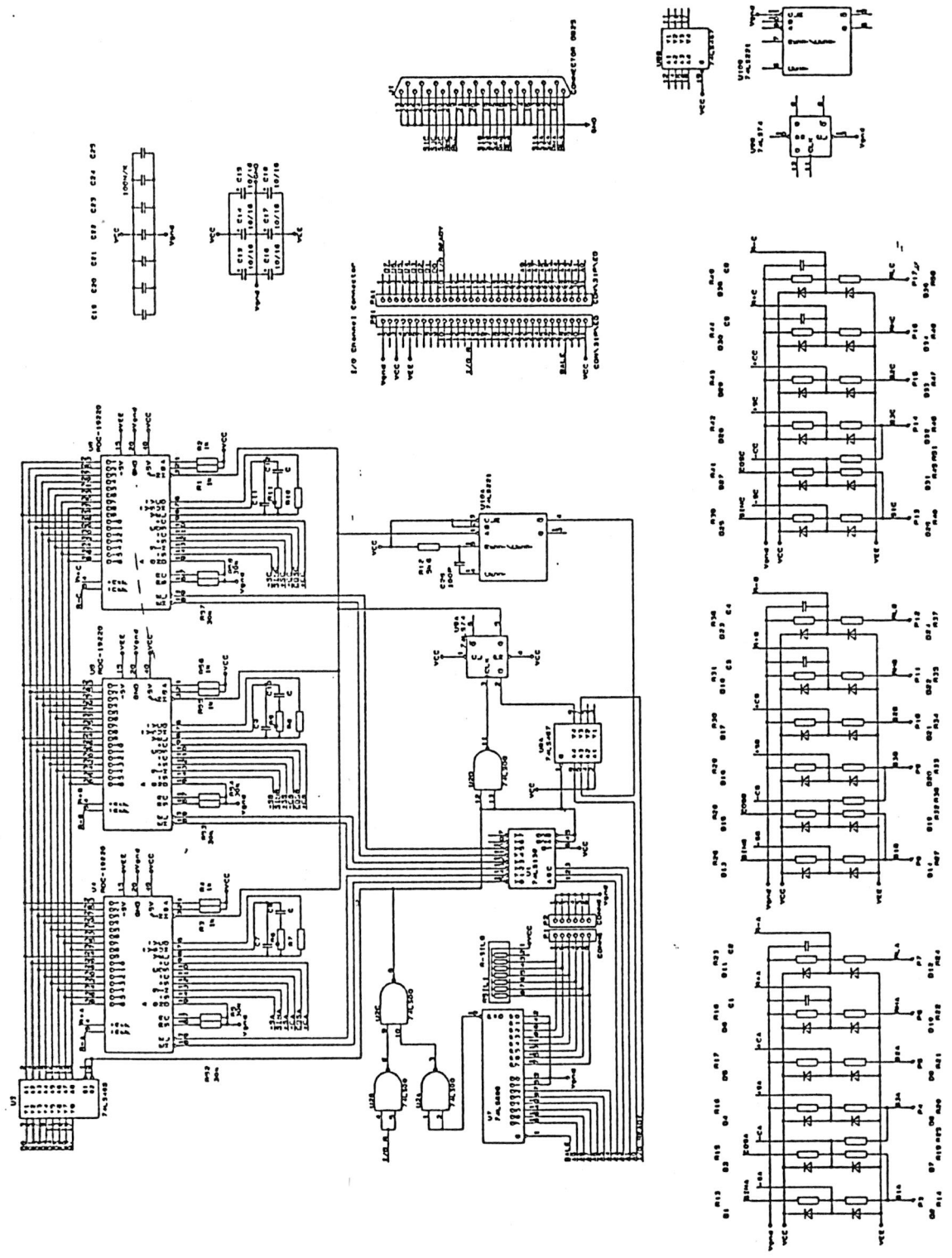


figure C.1. The complete card layout

FUNCTION TABLE

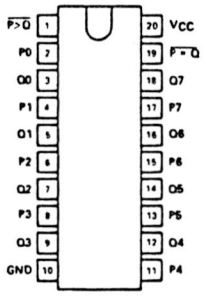
INPUTS			OUTPUTS	
DATA	ENABLES			
P, Q	$\overline{G}$, $\overline{G1}$	$\overline{G2}$	$\overline{P=Q}$	$\overline{P>Q}$
P = Q	L	L	L	H
P > Q	L	L	H	L
P < Q	L	L	H	H
X	H	H	H	H

H = high level, L = low level, X = irrelevant

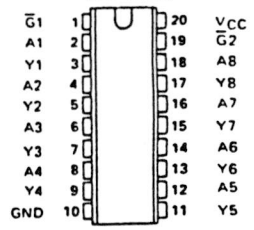
NOTES: 1. The last line of function table applies only to those devices having enable inputs, i.e., 'LS686 thru 'LS689.

2. The $\overline{P<Q}$ function can be generated by applying the $\overline{P=Q}$ and $\overline{P>Q}$ outputs to a 2-input NAND gate.

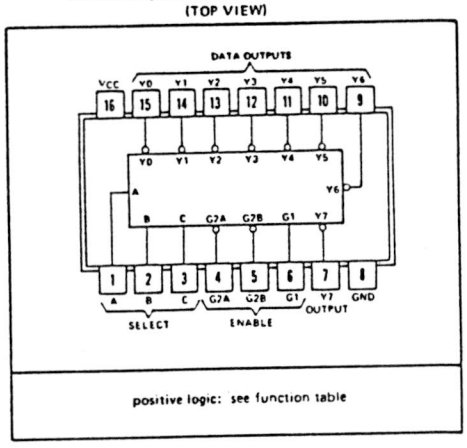
SN54LS688, SN54LS689 . . . J PACKAGE
SN74LS688, SN74LS689 . . . J OR N PACKAGE
(TOP VIEW)



SN54LS465 AND SN54LS466 . . J PACKAGE
SN74LS465 AND SN74LS466 . . J OR N PACKAGE
(TOP VIEW)



SN54LS138, SN54S138 . . . J OR W PACKAGE
SN74LS138, SN74S138 . . . J OR N PACKAGE
(TOP VIEW)



positive logic: see function table

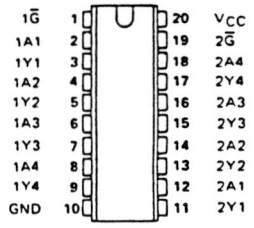
'LS138, 'S138
FUNCTION TABLE

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

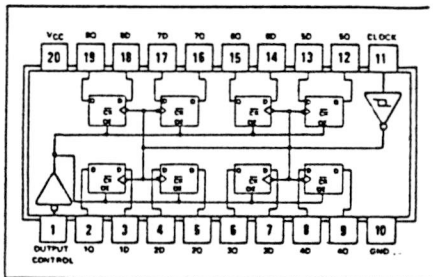
*G2 = G2A + G2B

H = high level, L = low level, X = irrelevant

SN54LS467 AND SN54LS468 . . J PACKAGE
SN74LS467 AND SN74LS468 . . J OR N PACKAGE
(TOP VIEW)

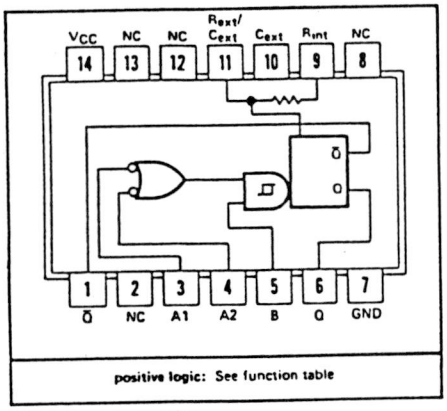


SN54LS374, SN54S374 . . . J PACKAGE
SN74LS374, SN74S374 . . . J OR N PACKAGE
(TOP VIEW)



logic: see function table

SN54121 . . . J OR W PACKAGE
SN74121 . . . J OR N PACKAGE



positive logic: See function table

NC—No internal connection

FUNCTION TABLE

INPUTS			OUTPUTS	
A1	A2	B	Q	$\overline{Q}$
L	X	H	L	H
X	L	H	L	H
X	X	L	L	H
H	H	X	L	H
H	↓	H	⊓	⊔
↓	H	H	⊓	⊔
↓	↓	H	⊓	⊔
L	X	↑	⊓	⊔
X	L	↑	⊓	⊔

For explanation of function table symbols, see page 3-8.

'LS374, 'S374
FUNCTION TABLE

OUTPUT CONTROL	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

See explanation of function tables on page 3-8.

figure C.2. The components and their function tables

Appendix D. The software of the card.

As discussed in chapter 4 there some software has been written to support the card. Two programs have been written, one to read the card and the second to use while testing. These programs are given on the next few pages. The programs will be discussed below.

Function SHIFT

This function has been implemented because there has been a production fault when making the card. The bits are read in reversed order by the computer, which means that bit 1 becomes bit 8, bit 2 becomes bit 7, etc. This function reverses the bits again so that the byte is correct again.

Procedure ROLL_PITCH_HEADING

When calling upon the procedure, one can get the angle of either roll, pitch or heading (depending on the value of the variable 'number'). It will also produce boolean variable called error, which will indicate if a error has occurred. For each synchro a different offset can be set, by changing the constants offset1, offset2 and offset3.

The error testing is done by reading the same byte two times and comparing them. With the least significant bit the difference may be one bit so that why a AND operation with 127 is performed. When you got two bytes which differ one bit you get and perform an AND operation, you get:

01100110	01100111
11111110	11111110
01100110	01100110

As can be seen, the two bit are the same so error flag will be set.

Program TEST

This program calculates the angle until a key is pressed and roll, pitch and heading can be obtained by respectively choosing 1, 2 or 3. The results are printed on the screen.

```
Page   1, listing of TEST.PAS, date is 01-05-93, file date is 01-05-93, size is 1417 bytes.

 1 PROGRAM test;
 2
 3 uses CRT;
 4
 5 VAR a, key     : integer;
 6     goed       : boolean;
 7     angle      : real;
 8
 9
10
11
12
13 (*************************************************************************)
14 (* This program was used for testing the card and it's function is to con- *)
15 (* stantly adress on of the card's convertors and printing the anlgles un- *)
16 (* til a key has been pressed.                                             *)
17 (*************************************************************************)
18
19
20 BEGIN
21  goed:=true;
22  WHILE goed DO
23  BEGIN
24     angle:=0;
25     READ(key);
26     CASE key OF
27        1: begin
28             repeat
29             begin
30               writeln('this is the value of synchro 1');
31               roll_pitch_heading(1,angle);
32               writeln(angle)
33             end;
34             until keypressed
35           end;
36        2: begin
37             repeat
38             begin
39               writeln('this is the value of synchro 2');
40               roll_pitch_heading(2,angle);
41               writeln(angle)
42             end;
43             until keypressed
44           end;
45        3: begin
46             repeat
47             begin
48               writeln('this is the value of synchro 3');
49               roll_pitch_heading(3,angle);
50               writeln(angle);
51             end;
52             until keypressed
53           end;
54        4: goed:=false
55
56     END;
57  END;
58 END.
59
```

Page 1, listing of TAAK4.PAS, date is 01-05-93, file date is 01-05-93, size is 4737 bytes.

```
 1 (*************************************************************************)
 2 (* This function replaces the bits, which means that bit 8 will switch   *)
 3 (* place with bit 1, bit 7 will switch place with bit 2, etc. This is    *)
 4 (* done by first determining the value of the bit and then multiply it   *)
 5 (* with the value of the place it should be on. The new value of the byte *)
 6 (* is obtained by adding all the separate value's.                       *)
 7 (*************************************************************************)
 8
 9
10 FUNCTION shift (old:BYTE):BYTE;
11
12 VAR  new, conversion      :   BYTE;
13      power, p, weight     :   INTEGER;
14
15 BEGIN
16    weight := 1;
17    power := 128;
18    new := 0;
19    shift := 0;
20    conversion := 0;
21    FOR p := 1 to 8 DO
22    BEGIN
23       new :=((old AND weight) * power) DIV weight;
24       conversion := conversion + new;
25       power := power DIV 2;
26       weight := weight * 2;
27    END;
28    shift := conversion;
29 END;
30
31 (*************************************************************************)
32 (* This procedure gives the value's of one of three synchro's. The dif- *)
33 (* ferent synchro's are selected by addressing the procedure with the   *)
34 (* numbers 1,2 or 3. It will reset the synchro's and it will read each  *)
35 (* byte 2 times, so that error tests can be performed. The bytes that   *)
36 (* are read first have to be shifted and this is done by the function   *)
37 (* shift. The hardware card is addressed by the addresses $300-$30F. The*)
38 (* stepsize is obtained by deviding 360 by (2^16 - 1).                  *)
39 (*************************************************************************)
40
41 PROCEDURE roll_pitch_heading (number:INTEGER;VAR angle:REAL;
42                                VAR error:BOOLEAN);
43
44 CONST
45   roll1    = $300;
46   roll2    = $308;
47   pitch1   = $304;
48   pitch2   = $30C;
49   heading1 = $302;
50   heading2 = $30A;
51   reset    = $30F;
52   stepsize = 5.493080245e-3;
53   offset1  = 26541;        ( these values for offset are random and have )
54   offset2  = 26541;        ( to be set when used in the plane            )
55   offset3  = 26541;
56
57 VAR
58   b                          :   INTEGER;
59   amount_of_steps            :   WORD;
60   lsb1, msb1, msb, lsb, rsb,
61   lsbc, msbc                 :   BYTE;
62
63 BEGIN
64   angle := 0;
65   amount_of_steps := 0;
66   rsb := PORT[reset];
67   CASE number OF
68   1: BEGIN
69        rsb  := PORT[reset];
70        msb1 := PORT[roll1];
71        lsb1 := PORT[roll2];
72        msbc := PORT[roll1];
73        lsbc := PORT[roll2]
74      END;
75   2: BEGIN
76        rsb  := PORT[reset];
77        msb1 := PORT[pitch1];
78        lsb1 := PORT[pitch2];
79        msbc := PORT[pitch1];
80        lsbc := PORT[pitch2]
81      END;
82   3: BEGIN
83        rsb  := PORT[reset];
84        msb1 := PORT[heading1];
85        lsb1 := PORT[heading2];
86        msbc := PORT[heading1];
87        lsbc := PORT[heading2]
88      END;
89   END;
90
91   (*************************************************************************)
92   (* Here the case statement is used again, because every synchro has     *)
93   (* got a different offset, which is selected below.                      *)
94   (*************************************************************************)
95
96   CASE number OF
97   1:BEGIN
98        lsb := shift(lsb1);
99        msb := shift(msb1);
100       amount_of_steps := (lsb + msb * 256) - offset1;
101       angle := amount_of_steps * stepsize;
102     END;
103  2:BEGIN
104       lsb := shift(lsb1);
105       msb := shift(msb1);
106       amount_of_steps := (lsb + msb * 256) - offset2;
107       angle := amount_of_steps * stepsize;
108    END;
109  3:BEGIN
110       lsb := shift(lsb1);
```

```
Page   2, listing of TAAK4.PAS, date is 01-05-93, file date is 01-05-93, size is 4737 bytes.

 111        msb := shift(msb1);
 112        amount_of_steps := (lsb + msb * 256) - offset3;
 113        angle := amount_of_steps * stepsize;
 114      END;
 115    END;
 116
 117
 118    {*************************************************************************}
 119    {* This is where the error-testing starts. First the values of the     *}
 120    {* bytes are tested on their correctness. Of the lsb1 only the 7 least *}
 121    {* significant bits are compared, because the 8th bit alters al the    *}
 122    {* time, this is due to the accuracy of the chip. Also the amount of   *}
 123    {* steps is checked as wel as the angle.                               *}
 124    {*************************************************************************}
 125
 126    lsbc := lsbc AND 127;
 127    lsb1 := lsb1 AND 127;
 128    IF (lsb1 = lsbc) AND (msb = msbc)
 129       THEN error := FALSE
 130       ELSE error := TRUE;
 131
 132    IF ((amount_of_steps < 0) OR (amount_of_steps > 65535))
 133       AND (error = FALSE)
 134    THEN error := FALSE
 135    ELSE error := TRUE;
 136
 137    IF ((angle < 0) OR (angle >= 360)) AND (error = FALSE)
 138    THEN error := FALSE
 139    ELSE error := TRUE;
 140
 141 END;
 142
 143
 144
 145
 146
```

Appendix E. Component value list

Synchro-to-PC converter; Partvalues.

c1,c2,c3,c4,c5,c6	1u (lowpass filter tegen spikes)
c7,c9,c11	4n7 (dynamica converter)
c8,c10,c12	47n (dynamica converter)
c13,c14,c15,c16,c17,c18	10u tantaal 16V (ontstoring voeding)
c19,c20,c21,c22,c23,c24,c25	100n keramic (ontstoring voeding)
c26	100p (delay tijd)
d1,d2,d3,d4,d5,d6,d7,d8, d9,d10,d11,d12,d13,d14,d15, d16,d17,d18,d19,d20,d21, d22,d23,d24,d25,d26,d27, d28,d29,d30,d31,d32,d33, d34,d35,d36	1N4448 (overspanningsbeveiliging)
r1,r2,r3,r4,r55,r56	1k 1/8W (pullup weerstanden)
r5,r52,r53,r54,r57,r58	30k 1/8W (max tracking rate en carrier freq)
r6,r9,r11	180k 1/8W (dynamica converter)
r7,r8,r10	47k 1/8W (dynamica converter)
r12	5k6 1/8W (delay tijd zie c16)
r13,r16,r26,r29,r39,r42	56k 1/8W precisie (spanningsdeler signaal)
r14,r19,r20,r25,r27,r32, r33,r38,r40,r45,r46,r51	330k 1/8W precisie (spanningsdeler signaal)
r15,r17,r28,r30,r41,r43	190k 1/8W precisie (spanningsdeler signaal)
r18,r23,r31,r36,r44,r49	8k2 1/8W (spanningsdeler Vref)
r21,r34,r47	165k 1/8W precisie (spanningsdeler signaal)
r22,r24,r35,r37,r48,r50	82k 1/8W (spanningsdeler Vref)
rsil1	pack 7 weerstanden elk 4k7 (pullup weerstanden ttl)
u1	74ls138
u2	74ls00
u3	74ls465
u7	74ls688
u8	74ls467
u9	74ls74
u10	74ls221