Boletim da Sociedade Astronômica Brasileira, **34**, no. 1, 257-261

Upgrade of frequency synthesis circuit of BDA

Brazilian Decimetric Array

T. Machado[1,2], K. Souza[1], C. Strauss[1], M. Abelha[1,3], & J. Cecatto[1]

[1] National Institute for Federal Research, Sao Jose dos Campos, SP, Brazil
[2] University of Vale do Paraíba, Sao Jose dos Campos, SP, Brazil
[3] Federal University of Minas Gerais, Belho Horizonte, MG, Brazil

Abstract. A radiotelescope receiver, for f >1GHz, usually is of the superheterodyne type to reduce the loss of signal power in the path through cables and electronic components to the data record. In turn, an interferometric radio receiver is considerably more complex and may involve other frequency conversion steps along its constituent parts for signal stability, mainly in terms of phase. As an interferometer is made up of many elements (antennas), with the frequency being adjusted at the receiver of each one individually, significant time is consumed in this process. In order to speed up the process of adjusting the operating frequency, to minimize losses and to maximize signal stability, the automation of the frequency synthesis of the Brazilian Decimetric Array (BDA) radiointerferometer receivers has been developed. This pioneering technological development employs a low-cost microcontroller along with conventional circuits. It is a technology developed within the institution itself, based on existing competence, and which implies a significant cost reduction in terms of material and mainly human resources. In this work, we present characteristics of the "hardware" of the automation system for the synthesis of frequencies of the BDA receivers, its adjustment and control parameters, the results of bench tests and the results of the validation tests in operational condition of observation with the antennas.

Resumo. Um receptor de radiotelescópio, para f >1GHz, geralmente é do tipo super-heteródino para reduzir a perda de potência do sinal no percurso por cabos e componentes eletrônicos até o registro dos dados. Por sua vez, um receptor radio interferométrico é consideravelmente mais complexo, podendo envolver outras etapas de conversão de frequência, ao longo de suas partes constituintes para a estabilidade do sinal, principalmente em termos da fase. Como um interferômetro é composto de muitos elementos (antenas), com a frequência sendo ajustada no receptor de cada uma individualmente, um tempo significativo é consumido nesse processo. Com o intuito de agilizar o processo de ajuste da frequência de operação, minimizar as perdas e maximizar a estabilidade do sinal, foi desenvolvida a automatização da síntese de frequência dos receptores do radiointerferômetro Brazilian Decimetric Array (BDA). Esse desenvolvimento tecnológico pioneiro emprega microcontrolador de baixo custo junto com circuitos convencionais. Trata-se de tecnologia desenvolvida dentro da própria instituição, a partir da competência existente, e que implica em significativa redução de custos em termos de recursos materiais e principalmente humanos. Neste trabalho, apresentamos características do "hardware" do sistema de automatização da síntese de frequências dos receptores do BDA, seus parâmetros de ajuste e controle, os resultados de testes de bancada e os resultados dos testes de validação em condição operacional de observação com as antenas.

Keywords. Instrumentation: interferometers – Sun: activity – Sun: radio radiation – Techniques: interferometric – Telescopes

1. Introduction

The instrument consists of 26 sets of parabolic dishes (4 meters in diameter) and 2 (two) servomotors (for elevation and for azimuth), in a “T” layout, 252 meters in East-West direction, and 162 meters in the South direction, as shown in Figure 1.

The positioning of each antenna is due to its panel, with a programmable logic controller (PLC), a driver for two axis and two servomotors. After more than a decade of development, some components may be facing their end of useful life, becoming obsolete or discontinued. The components that are no longer being produced includes the motor controller and the frequency synthesis controller, this second being the scope of this report.

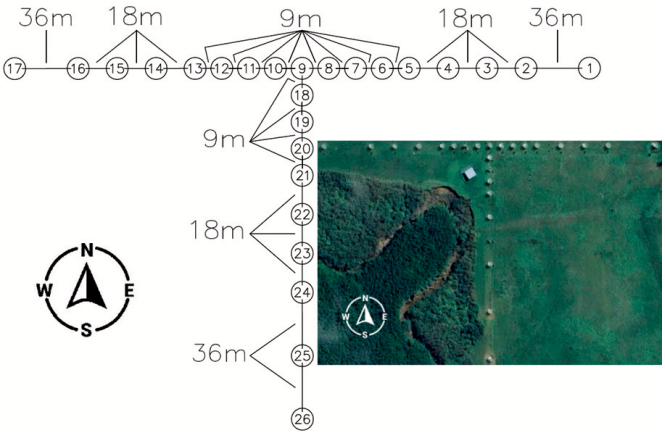

Figure 1. Layout and satellite photo of the instrument

2. Current situation

The modules that are responsible for the information provided to the Phase Locked Loop (PLL) for the control frequency synthesis of the drivers for the antennas are not working properly anymore, causing the interruption of the instrument operation. They had production and sale ceased, becoming unavailable for replacement. However, the frequency synthesis module and the *control unit* were not sufficiently detailed in the original project, presented in the Neuron report (RTE-3030-2000-XA) as a version of the receiver design. Then, the specialist team, headed by the Technician Khristhiano Souza and Technologist Dr. Cesar Strauss, under supervision of Researcher Dr. Roberto Cecatto, has made a detailed survey of the connection diagram and the signals measured at each point of board, as shown in Figure 2.

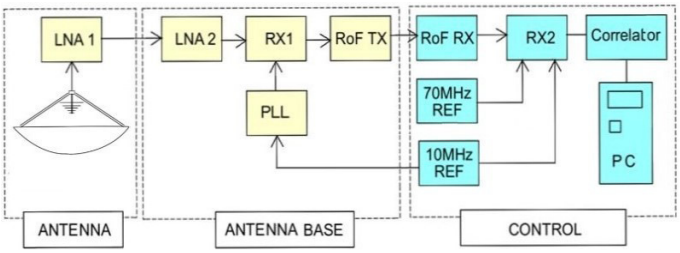

Figure 2. Instrument basic block diagram

3. Scope

The scope of this document includes the design of a new and modern circuit, using current automation, to the frequency synthesis modules for the signals that are to be observed, such the 1405MHz frequency. From this information, and having the minimum necessary parameters, sharp defined in the PLL integrated circuit specification catalog (Peregrine Semiconductor: PEE3336), and the updated diagram of the connectors and cables, the list of connections for the Arduino, and, consecutively, the software to be embedded has been developed and implemented.

4. Frequency synthesis

From a reference signal and using frequency synthesis methods, the frequency of signal in each receiver whose operational frequency vary into a wide range (1.2 – 1.7)GHz can be generated. The need to obtain signals with frequencies different from the reference signal is very important to telecommunications or signal transmission, for example. Frequency synthesis can be defined as a multiplication of the frequency, whereby the divisor that is inserted between the signal and the phase comparator is selected according to the user's wishes. These operations are important to the instrument because they allow information to be transmitted from the antenna panel to the receiver in the control room (where the monitoring and control software runs) and, vice versa, from the control room to the antennas, at the field. This information carries the status of the hardware telemetry that is accommodated in the Panel at the base of the antenna tower. Such information received by the unit are commands that, after conditioning in the unit itself, result in control voltages to turn on or off electronic circuits, change the state of switches or a "bit train" to adjust the frequency of local oscillators and the signal source. Also, into the panel, there is a source that generates the secondary voltages used by the units and electronic circuits. In general, this power supply has status telemetry of secondary currents and voltages and "ON/OFF" status of the power supply itself. They also have secondary voltage and current test points to allow the measurement of these quantities with external instrumentation in the case of fault diagnosis. This source has the facility to be remotely controlled by means of "on" and "off" from the control room. PLL is a control system that generates an output signal with instantaneous frequency and phase in sync with the sampled signal from an input signal. Once the output frequency remains "proportional" to the input frequency, such a system can track the frequency of the sampled signal, as well as generate multiple frequencies of the input signal. The properties of control loop allow the development of most varied applications, such as radio, telecommunications, signal analysis and others. A basic PLL system consists of a phase detect or phase comparator, a loop filter and a voltage controlled oscillator (VCO), as shown in Figure 5.

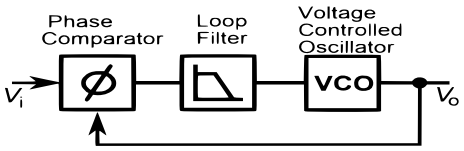

Figure 3. PLL basic block diagram

5. Phase-locked loop - PLL

The function of the phase detect or comparator block is to compare the sine of the input signal to the cosine of the VCO block in order to generate an error signal proportional to the angle error. To do that, the block in question multiplies the output signal of the VCO by the signal measured at the input.

However, the error information has a variable that is twice the input frequency, and this must be removed for the system works properly. This error is the input for the loop filter, which is a PI (proportional integral) control, used to reset the error, allowing the system to enter steady state. As stated earlier, the nonlinear component of the phase detect output must be removed. Than, a band-reject filter may be used to eliminate the linearity that is imposed by multiplying the main frequency by two.

6. Frequency synthesis update

The update predicts the replacement of the "Tower Base Computer" and "Tower Base Monitoring and Control Unit" by a single new system, fully developed by BDA team, consisting of an Arduino, which is a modern miniaturized board for prototyping and automations runs small embedded software and is extremely versatile, with very great expandability. This replacement is not only necessary, but also highly recommended, since the original components may, very soon, have problems and they already are difficult to be acquired. However, the documentation of the original project, regarding frequency synthesis and the "control unit", was not detailed enough to allow the correct development of a new equivalent and fully operational unit.

7. Arduino plataform

Arduino™ is a platform used for prototyping electronic and automation circuits. It consists of a microcontroller board and its programming environment. Both the hardware and the programming environment are open source, i. e., they can be modified and reproduced with no restrictions, enabling the most varied ranges of solutions.

7.1. Microcontrollers

A microcontroller is a family of integrated circuits that incorporates various functionalities. Some models were known as "single-chip computer", since they have the basic functions that were necessary for the assembly of an operational computer. These components are used in various applications of embedded systems, such as cars, appliances, airplanes, home automation and, even, in scientific instruments. The fundamental difference between microprocessor and microcontroller is that the second one needs auxiliary modules to be able to perform its tasks, such as memories (volatile and non-volatile), communication interfaces and even oscillators. On the other hand, the microcontroller has already in its internal architecture those elements and even more functions, depending on the model.

7.2. Embedded systems

These are microcontrolled digital systems for specific applications where the controller is attached to the system it controls. These systems can perform several sets of predefined tasks and are built into the system. Used in specific tasks, it makes possible to optimize certain equipments, with reduced sizes, requiring less computational resources, with reliability and reduced final cost. In general, the systems are used in applications with few requirements. They are usually designed from a well-defined application and for specific tasks. The system can still run with very limited computational resources, that is, with the absence of keyboards, screens and even with low memory.

8. Theoretical development

To observe signals at 1405MHz, with FI output at 70MHz, the PLL frequencies must be set to 2240MHz and to 905MHz. In this way, we have:

$f = 2240 - 905 + 70$

$f = 1405\text{MHz}$

To do that, according to the manual of the PLL integrated circuit "PEE3336", the required programming must follow the explanation below.

Observing the formula:

$$F_{in} = [10x(M+1) + A]x\frac{f_r}{(R+1)} \quad (1)$$

wich:

$A \leq M + 1$ And $1 \leq M \leq 511$

the adopted reference frequency is:

$f = 10\text{MHz}$

Thus, to the needed frequencies of 905MHz and 2240MHz, the parameters found will necessarily be the described on Table 1, shown below.

Table 1. Frequencies and parameters

F = 905 MHz		F = 2280 MHz
R = 19	And	R = 0
M = 180		M = 21

In order to the integrated circuit PEE3336 PLL generate the needed frequencies, the Arduino circuit must send the *R*, *M* and *A* parameters to the PLL in a serial channel. As described in Table 2. "Primary Register Programming", the information must be sent respecting both the *MSB* and *LSB* (respectively, Most Significant Bit and Least Significant Bit), with the *MSB* transmitted first and the *LSB* transmitted last. Due to it be a serial communication, data is transmitted one bit at a time, sequentially, on a given communication channel. A well-known format of serial communication is the Universal Serial Bus (USB) protocol. Then, the data must be received by the PLL integrated circuit in this format. This data must be transmitted according to the Timing Diagram shown in Figure 8.

Table 2. Primary register programming

Interface Mode	$\overline{Enh}$	$\overline{Bmode}$	Smode	R_5	R_4	M_8	M_7	$\overline{Pre\_en}$	M_6	M_5	M_4	M_3	M_2	M_1	M_0	R_3	R_2	R_1	R_0	A_3	A_2	A_1	A_0
Parallel	1	0	0	M2_WR rising edge load					M1_WR rising edge load							A_WR rising edge load							
				D_3	D_2	D_1	D_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
Serial*	1	0	1	B_0	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	B_{11}	B_{12}	B_{13}	B_{14}	B_{15}	B_{16}	B_{17}	B_{18}	B_{19}
Direct	1	1	X	0	0	0	0	$\overline{Pre\_en}$	M_6	M_5	M_4	M_3	M_2	M_1	M_0	R_3	R_2	R_1	R_0	A_3	A_2	A_1	A_0

MSB (first in) (last in) LSB

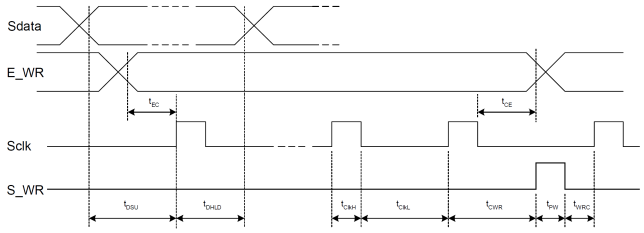

Figure 4. Timing diagram

9. Numerical development

The Table 2 shows that $0 \leq R \leq 63$ and:

1) *R* varies from *R*0 to *R*5, i. e., it has 6 bits:

0 to $(2^6 - 1)$, so from 0 to 63

2) *A* varies from *A*0 to *A*3, i. e., it has 4 bits:

0 to $(2^4 - 1)$, so from 0 to 15

3) *M* varies from *M*0 to *M*8, i. e., it has 9 bits:

0 to $(2^9 - 1)$, so from 0 to 511

Values outside these ranges are not allowed to be used in this programming. As the objective is to use the Arduino for these calculations, the simplest methods should be chosen. As equation 1, presented in item 8, has 3 (three) variables, the BDA team has developed a routine in Matlab™, whose main objective was to verify the possible values for the solution. With these values, it was possible to create the Table 3, that has sets of solutions that can be used in the development of the embedded software of the Frequency Synthesis Arduino. From the data obtained by the Matlab™ routine, the BDA team concluded that when we multiply the PLL output frequency by 2 (two), a visual pattern is verified and facilitates to obtain a formula that relates the *A* and *M* parameters to twice the output frequency. That is, the *tens* digit of this frequency is exactly the *A* parameter and the two digits of *thousands* and *hundreds*, subtracting one unit, gives the *M* parameter. This brings us closer to find a formula for *A* and *M*. The *R* parameter returns the smallest values for *A* and *M* parameters when its value is equal to 1 (i. e., $R = 1$). In addition, *R* parameter equals to 1 allows the visualization of the pattern described above. With a more simplified standard, the computational cost is also much lower, which becomes an excellent advantage to the simplicity of the chosen system (Arduino). From the observed pattern, the *module* function can be used in the value of the Output Frequency multiplied by 2 and then divided by 10 (the least significant digit can be

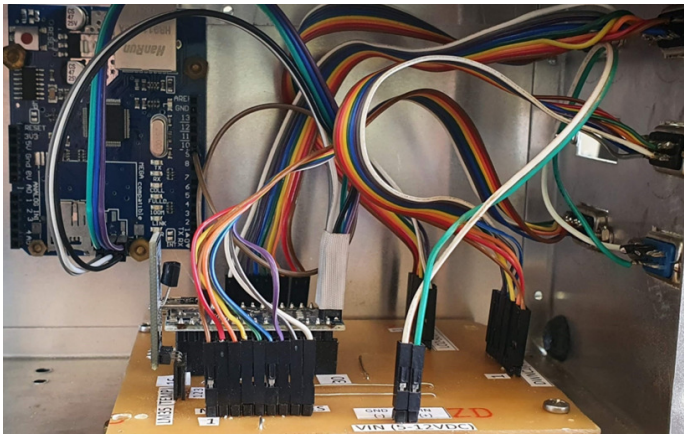

Figure 5. Prototype

discarded to find the desired parameters).

Exampling:

a) Output frequency = 2040MHz

$$\frac{Fx2}{10} = \frac{4080}{10} = 408$$

(whole division by 10 effects the same of shifting the digit from *tens* to *ones*).

The A parameter is found in the remainder of the division of 408 by 10.

$A = 408\%10 = 8$ (% represents the Modulus Function).

The M parameter is just the integer value of the quotient of dividing 408 by 10 subtracting 1 at the end. So, in this example:

$M = (408//10) - 1 = 39$ (// represents Integer Division).

b) Output Frequency = 2540MHz

$$\frac{Fx2}{10} = \frac{5080}{10} = 508$$

$A = 508\%10 = 8$

$M = (508//10) - 1 = 49$

10. Signals obtained

Based on the values found, a software was developed for the Arduino based prototype (Figure 10) to generate the pulses exactly the way the PLL integrated circuit needs to synthesize the frequencies.

The accuracy of these signals was monitored by the oscilloscope, as shown in figures 6 and 7.

References

Peregrine Semiconductor, “PEE3336: 3000 MHz UltraCMOS™ Integer-N PLL for Low Phase"

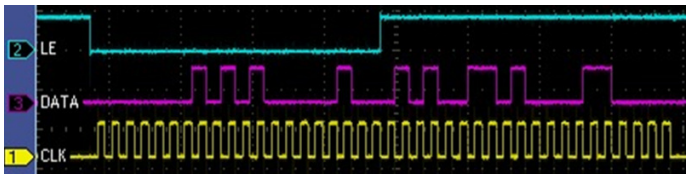

Figure 6. Timing diagram synthetized (1405MHz)

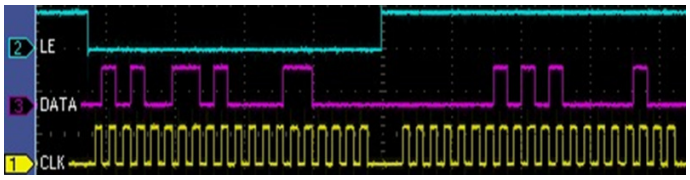

Figure 7. Timing diagram synthetized (1705MHz)

Appendix A: Routine in Matlab

The following routine was developed to test the possible values for A, M and R parameters.

Matlab Routine

```
fr = 10;
x = 1;
i = 1;
n = 1;
for f = 1205:5:1705
    Fin = f-70+905;
    for m = 1:1:511
      for a = 0:1:9
         for r = 1:1:63
             F = (10*(m+1)+a)*(fr/(r+1));
             if F==Fin
                A(i)=a; M(i)=m; R(i)=r; i=i+1;
             end
          end
       end
       end
       i = 1;
       S(n,:) = [Fin f A(x) M(x) R(x)];
       n=n+1;
end
S
```

Appendix B: A, M and R parameters

The routine presented in Section A allowed the creation of Table B.1, in which the value for the R parameter was conveniently choosen as 1. In the simulations, the routine had so much lower computational cost as smaller the value of R was. Then, the adopted value was $R = 1$. As there are several sets of A, M and R parameters that may be solutions, only the first solutions for a given frequency is presented. In order to increase the performance, the program must be the simpler possible in computing code. In that way, the first adopted value was $R = 1$. The next steps envolved the visual pattern that has been verified to obtain the simplest formula possible, as described in the Section 9 - "Numerical development".

Table B.1. *A* and *M* parameters (with $R = 1$)

F	f	F x 2	A	M		F	f	F x 2	A	M
2040	1205	4080	8	39		2295	1460	4590	9	44
2045	1210	4090	9	39		2300	1465	4600	0	45
2050	1215	4100	0	40		2305	1470	4610	1	45
2055	1220	4110	1	40		2310	1475	4620	2	45
2060	1225	4120	2	40		2315	1480	4630	3	45
2065	1230	4130	3	40		2320	1485	4640	4	45
2070	1235	4140	4	40		2325	1490	4650	5	45
2075	1240	4150	5	40		2330	1495	4660	6	45
2080	1245	4160	6	40		2335	1500	4670	7	45
2085	1250	4170	7	40		2340	1505	4680	8	45
2090	1255	4180	8	40		2345	1510	4690	9	45
2095	1260	4190	9	40		2350	1515	4700	0	46
2100	1265	4200	0	41		2355	1520	4710	1	46
2105	1270	4210	1	41		2360	1525	4720	2	46
2110	1275	4220	2	41		2365	1530	4730	3	46
2115	1280	4230	3	41		2370	1535	4740	4	46
2120	1285	4240	4	41		2375	1540	4750	5	46
2125	1290	4250	5	41		2380	1545	4760	6	46
2130	1295	4260	6	41		2385	1550	4770	7	46
2135	1300	4270	7	41		2390	1555	4780	8	46
2140	1305	4280	8	41		2395	1560	4790	9	46
2145	1310	4290	9	41		2400	1565	4800	0	47
2150	1315	4300	0	42		2405	1570	4810	1	47
2155	1320	4310	1	42		2410	1575	4820	2	47
2160	1325	4320	2	42		2415	1580	4830	3	47
2165	1330	4330	3	42		2420	1585	4840	4	47
2170	1335	4340	4	42		2425	1590	4850	5	47
2175	1340	4350	5	42		2430	1595	4860	6	47
2180	1345	4360	6	42		2435	1600	4870	7	47
2185	1350	4370	7	42		2440	1605	4880	8	47
2190	1355	4380	8	42		2445	1610	4890	9	47
2195	1360	4390	9	42		2450	1615	4900	0	48
2200	1365	4400	0	43		2455	1620	4910	1	48
2205	1370	4410	1	43		2460	1625	4920	2	48
2210	1375	4420	2	43		2465	1630	4930	3	48
2215	1380	4430	3	43		2470	1635	4940	4	48
2220	1385	4440	4	43		2475	1640	4950	5	48
2225	1390	4450	5	43		2480	1645	4960	6	48
2230	1395	4460	6	43		2485	1650	4970	7	48
2235	1400	4470	7	43		2490	1655	4980	8	48
2240	1405	4480	8	43		2495	1660	4990	9	48
2245	1410	4490	9	43		2500	1665	5000	0	49
2250	1415	4500	0	44		2505	1670	5010	1	49
2255	1420	4510	1	44		2510	1675	5020	2	49
2260	1425	4520	2	44		2515	1680	5030	3	49
2265	1430	4530	3	44		2520	1685	5040	4	49
2270	1435	4540	4	44		2525	1690	5050	5	49
2275	1440	4550	5	44		2530	1695	5060	6	49
2280	1445	4560	6	44		2535	1700	5070	7	49
2285	1450	4570	7	44		2540	1705	5080	8	49
2290	1455	4580	8	44						

Acknowledgements. Our acknowledgements are due to the financial agencies, to know, FAPESP, CNPq, FINEP, CAPES for all kinds of support to the project. We would also like to thank INPE for all the conditions given for the accomplishment this work, and to the colleagues at INPE, professors, researchers, advisors and, especially. the BDA Team, mainly represented by:

Telmo C. Machado	e-mail: telmo.machado@gmail.com
Khristhiano L. R. Souza	e-mail: khristhiano.souza@inpe.br
Cesar Strauss	e-mail: cesar.strauss@inpe.br
Matheus C. Abelha	e-mail: abelhamatheus@gmail.com
J. R. Cecatto	e-mail: jr.cecatto@inpe.br

Finally, I´d like to declare my appreciations and recognitions to everyone who, directly or indirectly, had their importance in the composition of this project.