4.2 IMPLEMENTATION LABVIEW

4.2.1 LabVIEW features

LabVIEW (short for Laboratory Virtual Instrument Engineering Workbench) originally released for the Apple Macintosh in 1986. It is a highly productive development environment that engineers and scientists use for graphical programming named "G" and unprecedented hardware integration to rapidly design and deploy measurement and control systems. LabVIEW provides tools to solve today's problems—and the capacity for future innovation—faster and more effectively.

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data, as well as tools for debugging the code. [35]

4.2.2 Program Sun Tracking

As is mentioned before, the software that calculates the position of the sun is already done. So, it has to be implemented by another program which tells to the dish controller to go to this position. To program this, first of all, do a diagram is very important to have a clear structure.

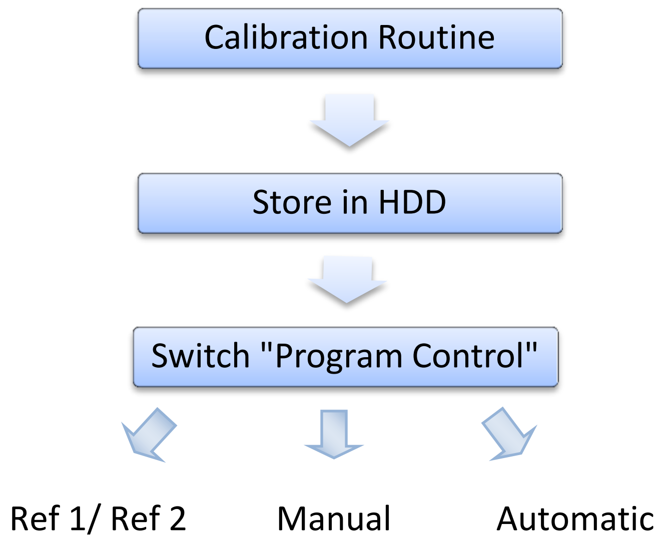

Figure 4.16 Overview Autotracking Dish-Stirling Solar Power Plant

The first step is to build the program apt for the calibration of the plant. This part of the program makes possible the transformation between the values given by LabVIEW that refers to revolutions of the motor and the inclination and rotation angle. The best way to be more precise is doing the measures experimentally, that is; the values of the main program are increased progressively, at the same time the angles of the dish are measured. The switch must be pushed every time that a value has to be saved. This calibration makes possible to do the transformation between the values of the program and the angles of the sun.

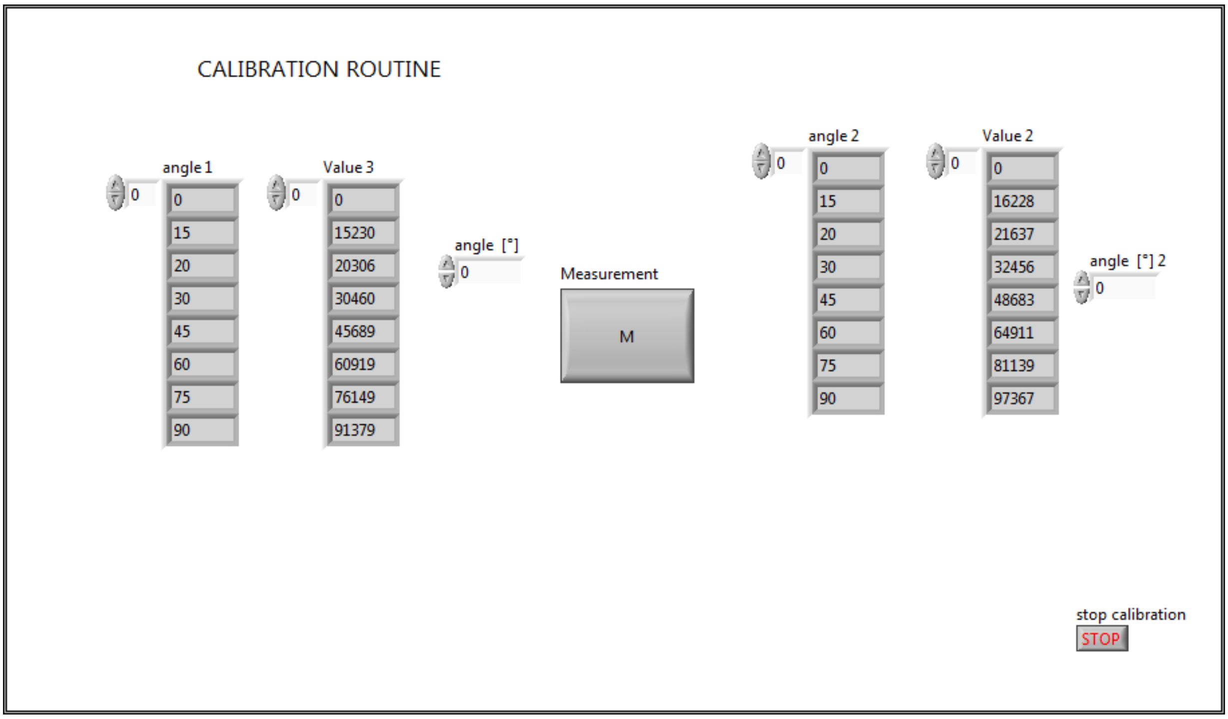

Figure 4.17 Calibration Routine

It is obvious that the calibration cannot be done every time that the program is opened, for this reason, for the second step, a configuration file is needed. This means that the information has to be stored in a hard disk drive[1] (HDD) and every time the program is executed, these values have to be recovered. This step is automatically. The Configuration File Vis can be used on any platform to read and write files created by the Vis.

[1] Hard disk drive (HDD): is a data storage device used for storing and retrieving digital information using rapidly rotating discs coated with magnetic material. An HDD retains its data even when powered off.

So, these two previous steps are only necessary do once. Then, the program is controlled by the switch "Program Control". Before to start the program it is in the option "Idle" that means not in operation.

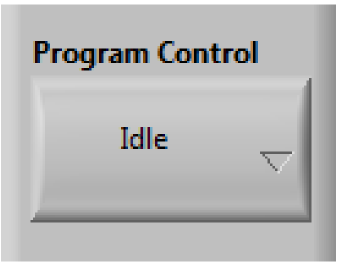

Figure 4.18 Switch "Program Control"

This switch has different options to choose. The first command when the program starts to run is the option "Ref 1". This option means that the motor 0 goes to the reference search of the switch. It is also the reference position of the dish. Automatically, the second option in the button, "Ref 2", is also executed, so, the motor 1 goes to the reference position.

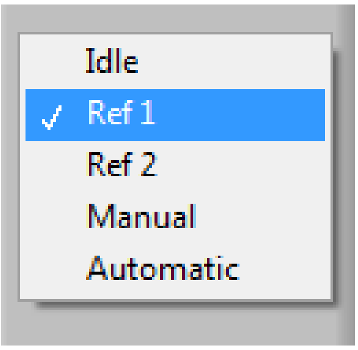

Figure 4.19 Options "Program Control"

After the plant is in his reference position, it is possible to activate the mode "Manual", where the instructions can be executed manually; or the mode "Automatic" that means the plant goes to the sun position and it tracks the sun with the previous calibration and the part of automatic control.

When the option "Automatic" is activated, a graphic representation of the functions is created. With the curve fitting[2] is possible to get the slope and the offset. These two values (two for each axis) are written in a file that is read later on.

---

[2] Curve fitting: is the process of constructing a curve or mathematical function that has the best fit to a series of data points, possibly subject to constraints.

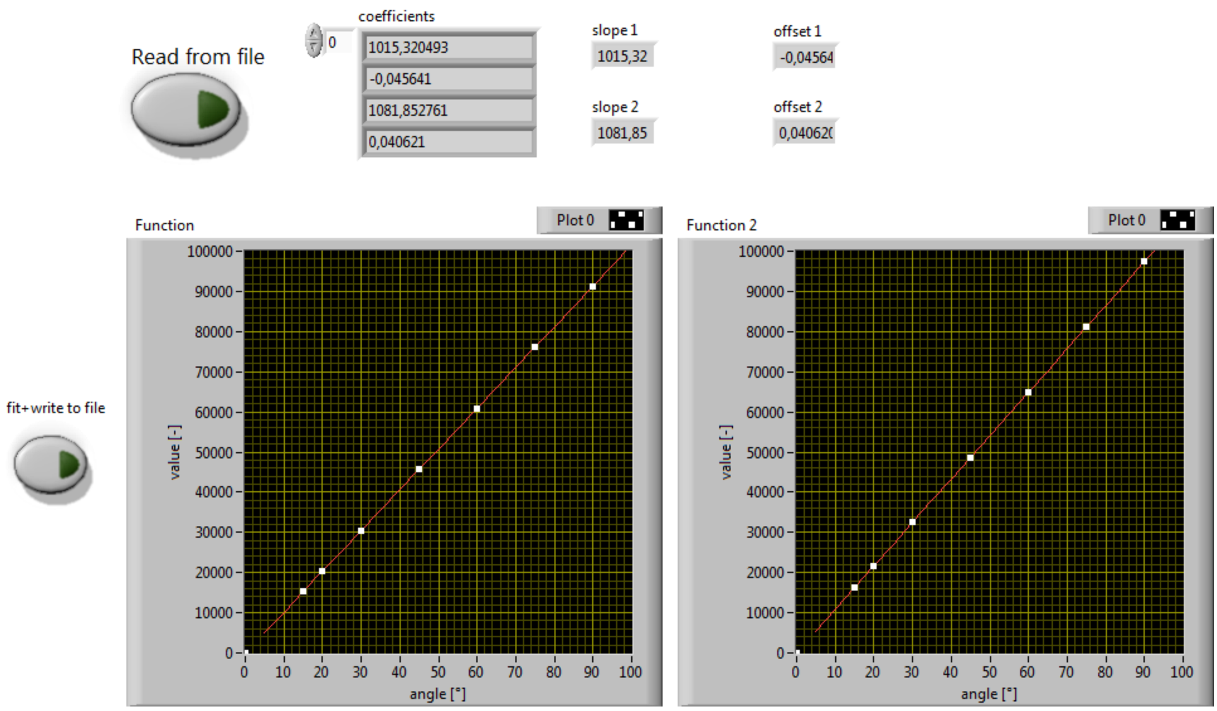

Figure 4.20 Configuration file and curve fitting

The data are shown in the "coefficients" with the following order: slope 1 and offset 1 (from the horizontal axis, azimuth) and slope 2 and offset 2 (from the vertical axis, altitude). Next to the coefficients there are the same values in order to check them.

Together with this part, there is another principal part that is needed because the "Automatic" mode works. This part was made by another student in his bachelor thesis [11]. This software uses the current time to estimate the sun position with the solar azimuth[3] and the solar altitude[4]. Both angles are represented in the following figure:

[3] Solar azimuth: is the azimuth angle of the sun. It defines in which direction the sun is. Is the compass direction from which the sunlight is coming.

[4] Solar altitude: the angle between the horizon and the centre of the sun's disc.

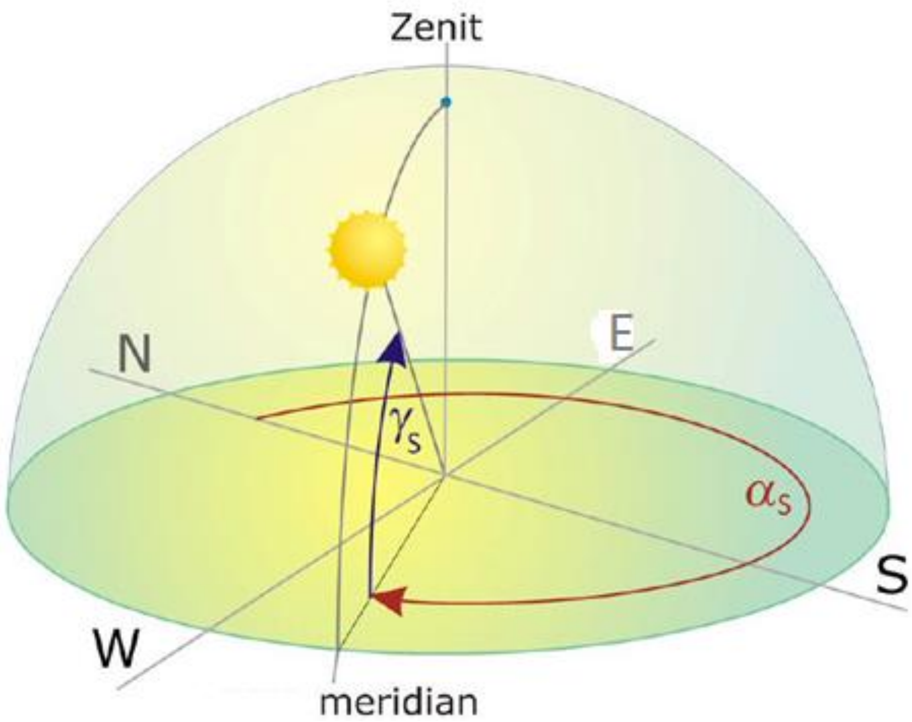

Figure 4.21 Sun charts [11]

This is the appearance of the automatic control part:

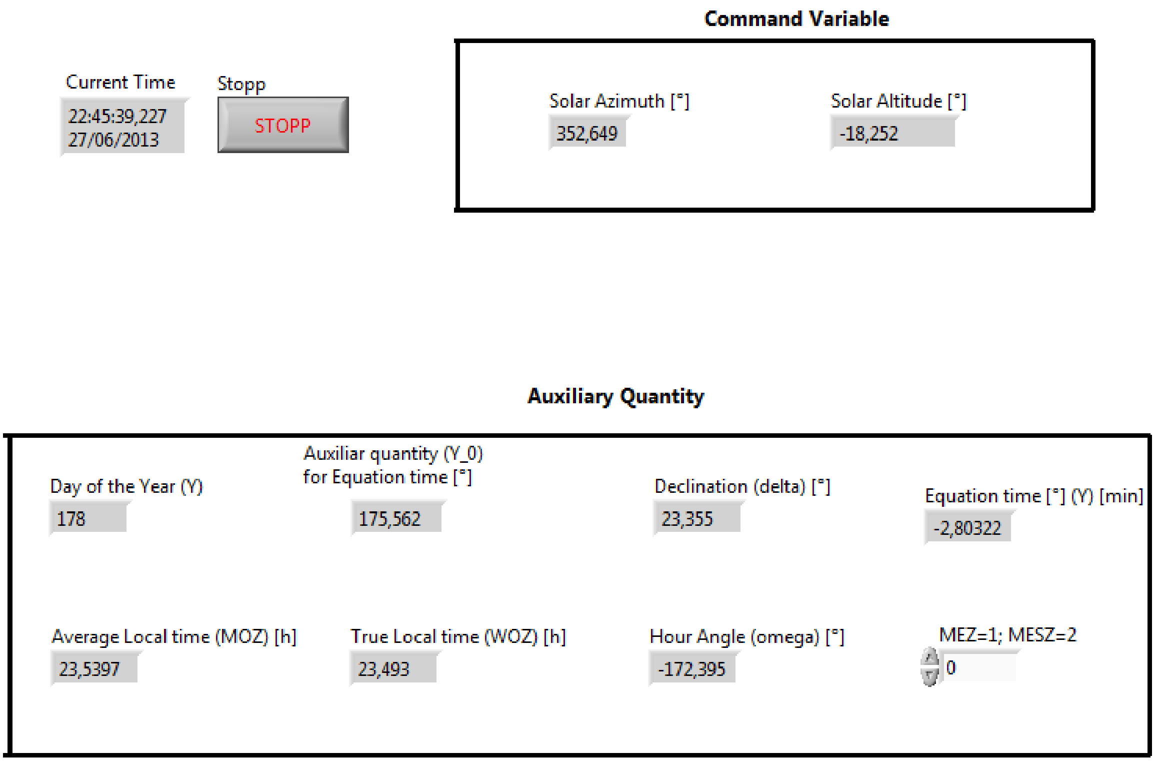

Figure 4.22 Calculation Sun position [11]

There are the two mentioned angles at the top. They are very important for the later calculation because of they are needed to know the corresponding values of the motors. Also the current time and date is shown. The other values below are dispensable to run the program. The Declination Angle (δ) defines the angle between the center of the Sun and the celestial equator[5], and this corresponds to the inclination of the Earth. The maximum and minimum declination of the sun over the year, expressed in decimal units is: +23.45° and -23.45°. The equation of the time is the difference between apparent solar time[6] and mean solar time[7]; at any instant, this difference will be the same for every observer on Earth. The Hour Angle (ω) is used to calculate the altitude of the sun and is counted from the respective meridian, positively in the afternoon and negatively in the morning. Is the angle between the half plane determined by the Earth's axis and the zenith (half of the meridian plane) and the half plane determined by the Earth's axis and the given point. The angle is with minus sign if the point is eastward of the meridian plane and with the plus sign if the point is westward of the meridian plane. [11]

This part in the Block Diagram is in a same loop called "Calculation of sun position". It can be updated every certain milliseconds.

The other mentioned choice "Manual" has three types of instructions:

- Reference Search (RFS): if this option is activated, the motor starts to run and doesn't stop until it hits the switch. The switches indicate the reference position of the two axes and the values start from zero.
- Move to position (MVP): this choice gives the possibility to introduce a certain value and the motor goes to this position.
- Motor stop (MST): this instruction stops immediately the selected motor but keeps the value where the motor was.

There is also the option to choose which motor is working, so the instruction only concerns the selected engine. Another possibility is contemplated about the type of value: absolute, relative or coordinate value. However, the calibration was done in absolute value, so, for the right operation of the program, the best way is to use this type.

[5] Celestial equator: is a great circle on the imaginary celestial sphere, in the same plane as the Earth's equator.
[6] Apparent solar time: is the time based on the rotation of the Earth with respect to the Sun.
[7] Mean solar time: is the hour angle of the mean Sun plus 12 hours.

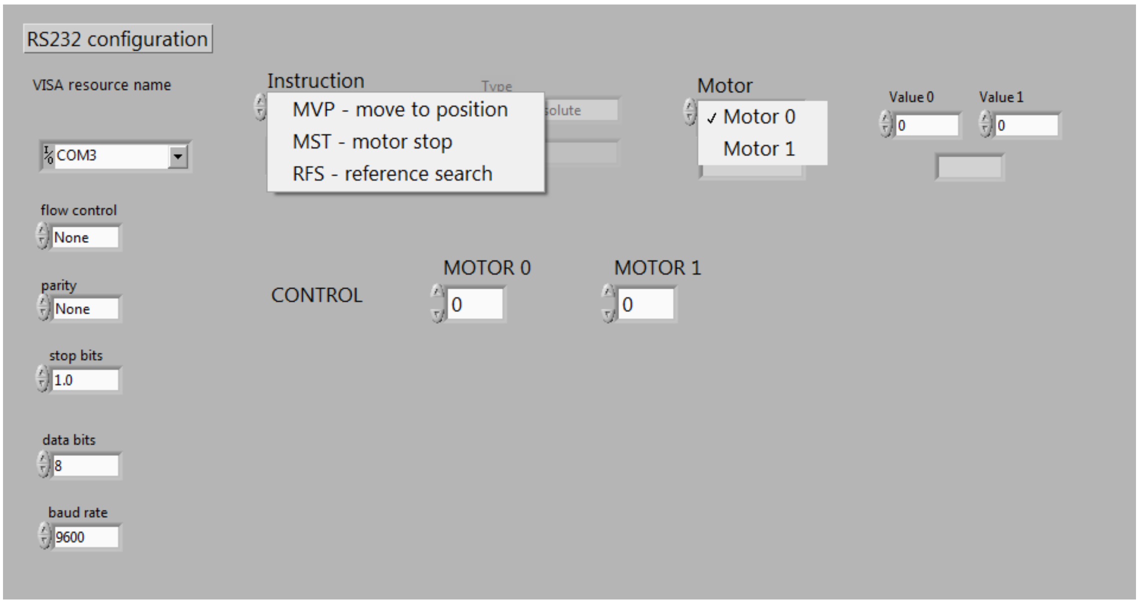

Figure 4.23 Manual Control

In the left side the interface is configured, in this case, the USB-485. In the “CONTROL” the desired values can be changed and these appear in the box “Value 0” or “Value 1”.

To write these commands it is important to learn about another software called TMCL-IDE (TrinamicMotionControlLanguage TMCL™ Integrated Development Environment). The board comes with the preinstalled firmware. The integrated development environment TMCL-IDE for PC can be downloaded and used free of charge. It is possible to remote control the stepRocker™ or to write and download complete command sequences for stand-alone use. [28]

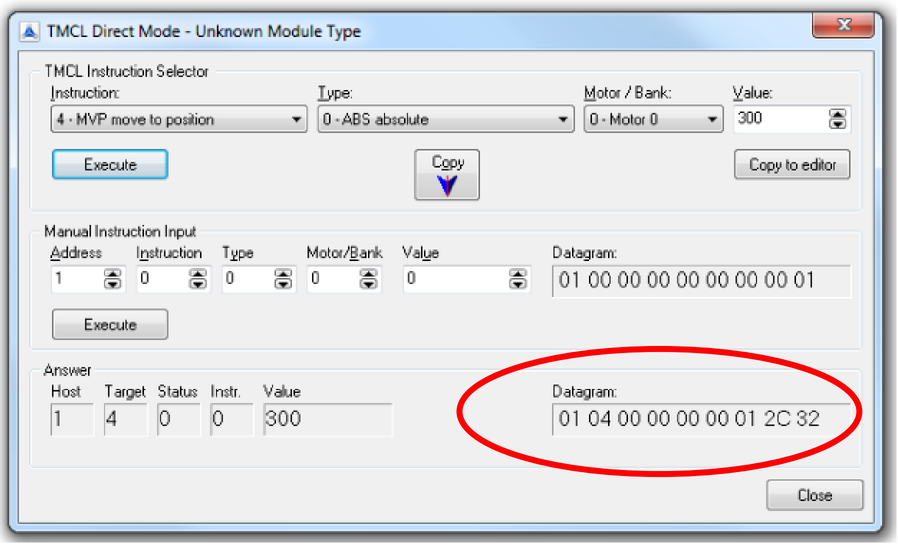

Figure 4.24 TMCL-IDE Application

Nonetheless, in this project, LabVIEW is used because the structure is more complicated and it has to work automatically. This software generates a code (Hexadecimal code) which means the instruction that has to be executed for the board and, to carry out the same function in LabVIEW is necessary to generate the same code. The following table shows the string for each instruction (the red numbers mean that are variables, depend on the number of the value):

Command	Description		Data to write
MVP	Move to position	ABS- Absolute	0104000000006590FA
		REL- Relative	0104010000006590FB
		COOR- Coordinate	0104020000006590FC
MST	Motor stop		010300000000000004
RFS	Reference search		010D0000000000000E

Table 4.5 Communication between TMCL and LabVIEW

To generate these commands is important to know the meaning of each. For example, in case of the instruction "Reference Search":

Module address (2 digits) + Instruction (2 digits)	Type (2 digits) + Motor (2 digits)	Constant (4 digits)	Value (4 digits)	CRC (2 digits)
010D	0000	0000	0000	0E

Table 4.6 Example Hexadecimal code

The way to generate in LabVIEW these commands is shown in the following example. This is the case of the motor 0 when the option "Ref. 1" is activated and the motor goes automatically to the reference position.

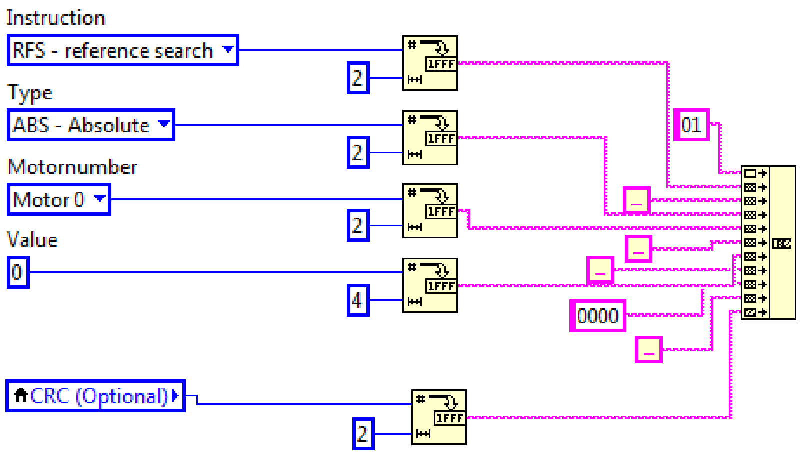

Figure 4.25 Block Diagram go to ref.1

The Front Panel of the program is organized like a folder and every tab contain one part of the program. A button “Quit LabVIEW program” is created to stop the software.

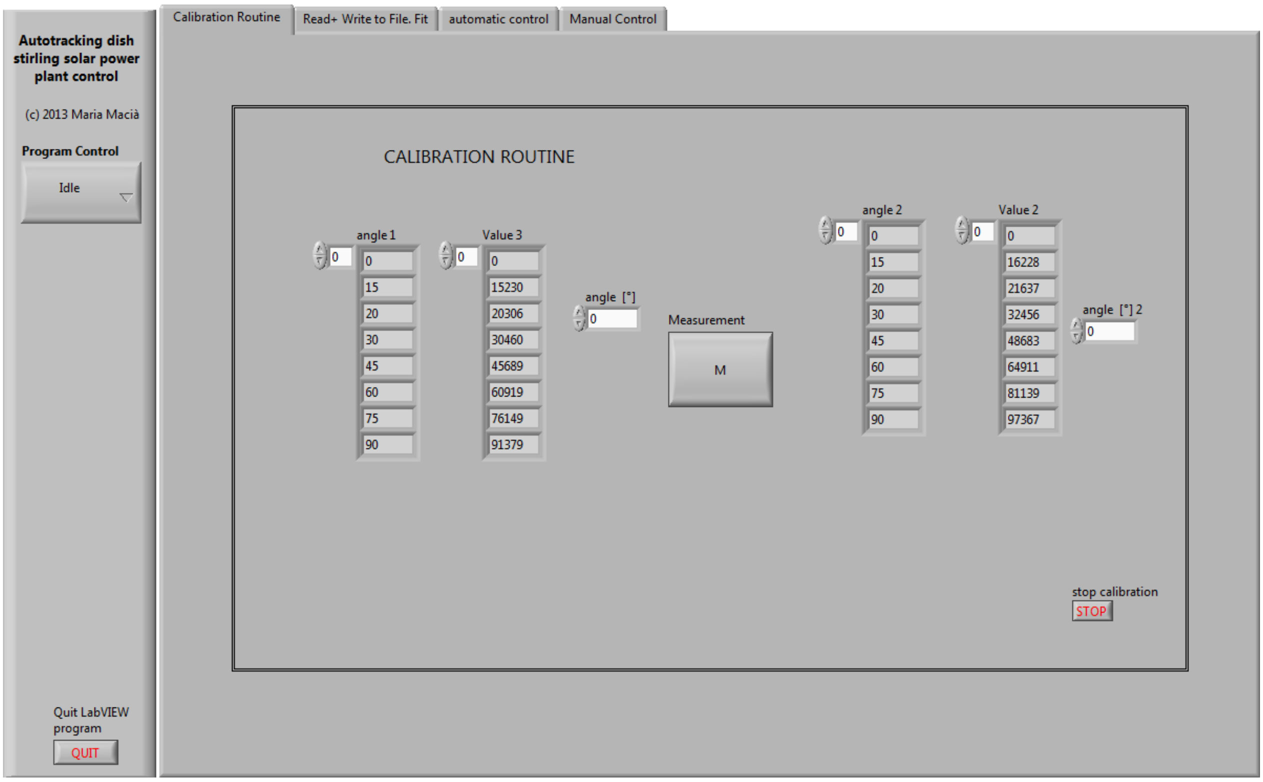

Figure 4.26 Autotracking Dish-Stirling Solar Power Plant

The view of the Block Diagram is structured in four loops. The first one, called “Manual control”, is where the main button “Program Control” with their options is programmed. The second, “Command sending structure”, is an automatic loop which sends constantly the movements to the board. The third one, “Calculation of sun position”, calculates the position of the sun and translates these degrees to the corresponding values. The last one, “Calibration Routine”, makes the calibration when is needed.

1. RESULTS

After all dedication to build the solar power plant, and primarily, the software for the automatic sun tracking, in the last days of the study something came up with the boards. A wrong clip was used to connect the motors to the Steprocker and it caused a short-circuit. The chips were removed as quickly as the mistake was seen. Even so, the boards were already damaged. They worked for some hours more, but, finally they burned.

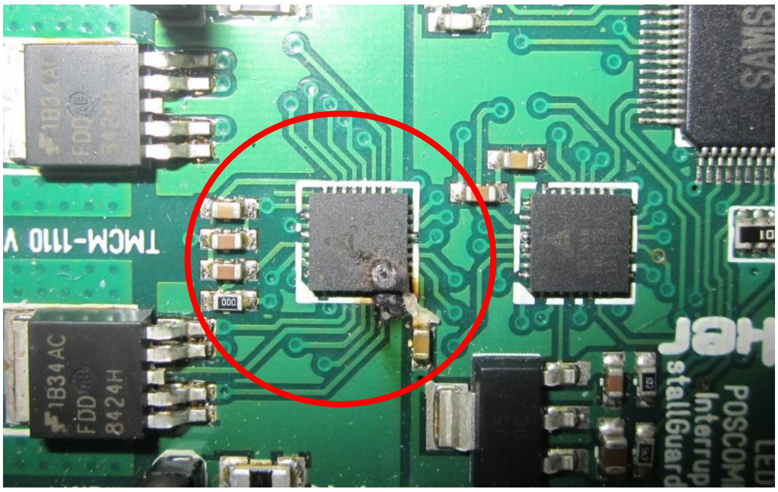

Figure 5.1 StepRocker damaged

For this reason, the part of "Calibration Routine" in the developed software was not able to be calibrated experimentally. The alternative solution was to calculate the points (the values which corresponding to every angle) to build the curve fitting with the gear ratio[8].

$$n_1 * r_1 = n_2 * r_2 \qquad 5.1$$

So, if the radius are known (r_1: radius of the motor; r_2: radius of the axis), it is possible to know the number of revolutions of the axis for a determinate revolutions of the motor. Moreover, it could be tested that one revolution of the motor is 53.000 in the value of the program. With this information, is easy to create a relation between the angle of the dish and the value in the program.

AXIS 1 (AZIMUTH)

[8] Gear ratio: is the ratio of the angular velocity of the input gear to the angular velocity of the output gear.

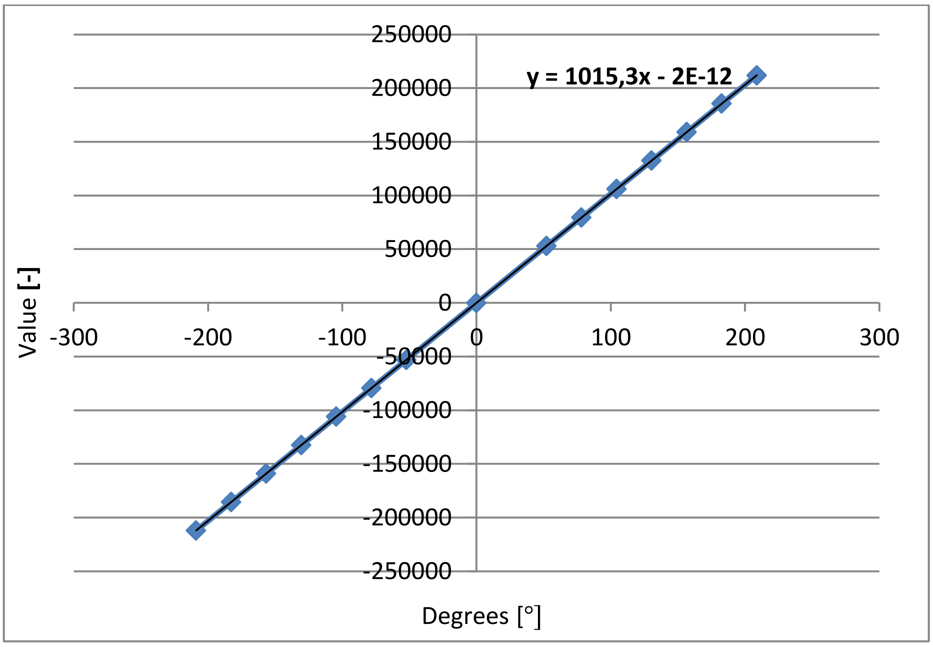

Figure 5.2 Graph curve fitting Axis 1

The equation of the function is shown in the graph, but more precise and with the magnitudes is the following:

$$n = 1015.326\alpha \qquad 5.2$$

Where:

"n" is the value in the software that represents the revolutions of the motors (one-dimensional)

α is the angle of the position of the dish [°]

The table with the correspondence degrees-values:

α	n
15	15230
20	20306
30	30460
45	45689
60	60919
75	76149
90	91379

Table 5.1 Relation degrees-value (Axis 1)

AXIS 2 (ALTITUDE)

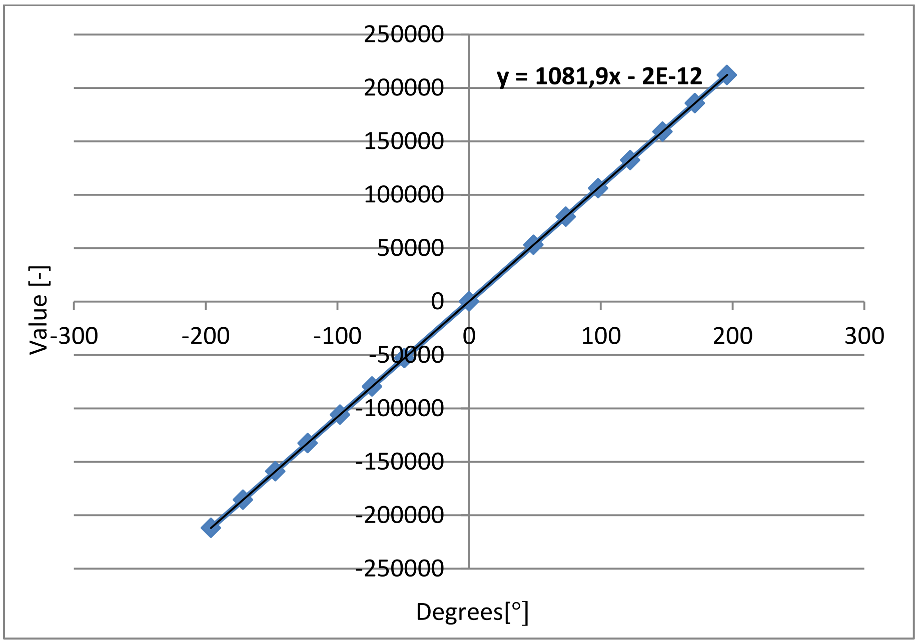

Figure 5.3 Graph curve fitting Axis 2

The equation of the function is shown in the graph, but more precise and with the magnitudes is the following:

$$n=1081.853\alpha \qquad 5.3$$

Where:

"n" is the value in the software that represents the revolutions of the motors (one-dimensional)

α is the angle of the position of the dish [°]

The table with the correspondence degrees-values:

α [°]	N [-]
15	16228
20	21637
30	32456
45	48683
60	64911
75	81139
90	97367

Table 5.2 Relation degrees-value (Axis 2)

So the transformation between the degrees to the value is as easy as solve the equation for each axis. The software reads the coefficients from the file (slope and offset) and answers the function with the angles of the sun.

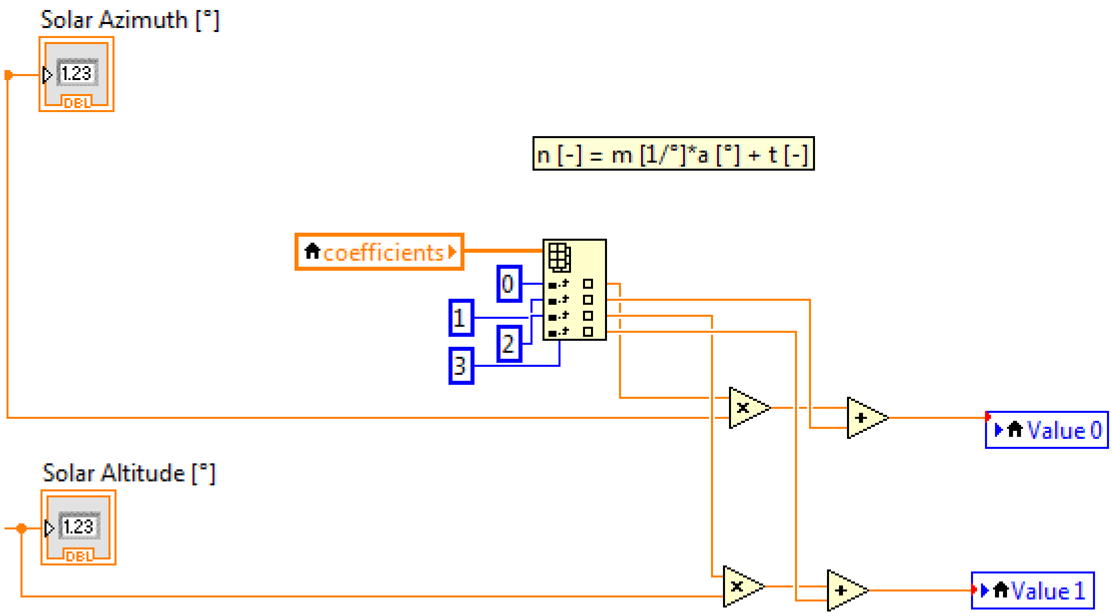

Figure 5.6 Translation between angles of the sun and values

Another issue to keep in mind is the mechanical part from the power plant, about the Stirling engine. The temperature of the motor got too hot after five minutes running, for this reason, build a cooler was necessary. The cooler consist in a cylindrical piece that covers the motor and for one hole the water comes in and for the other hole it comes out.

Figure 5.7 Cooler

2. CONCLUSIONS

As stated in the abstract, the main aim to achieve through this bachelor thesis was demonstrate the function of a Dish Stirling Solar Power Plant in order to show to the other students and visitors of the Bundeswehr University its operation. For that, a previous project was used as a base of this thesis. One of the most important factors was the torque of the motor because it was fundamental to buy the right motors. After that, it was necessary to look for the best one which met the criteria and it was not very expensive. A control boards that fits with the chosen engines were bought. All the connections was analyzed and checked. Moreover, the software from the existing project that indicated the position of the sun was implemented with new software. The last one measured the angles from the dish and both of them were agreed to tracking the sun.

Therefore, looking at the results, the initial objectives were fulfilled, although some problems that appeared the last week of the deadline. A mistake with the connections of the motors caused some damage to the boards and, unfortunately, the solar power plant could not run. However, the software for the sun tracking is already done. It was the main part of the work. It is explained in this report and also the software is attached with the documentation.

Along the entire project, I have learnt a lot about all the equipment I have used, especially about programming with LabVIEW that will be very useful for me in my future and about electronic. Additionally, I have also learnt about the difficulties that an engineer can find during a project and how to overcome it. I have realized that some unexpected events can appear and is required to look for a better way to get over the problems.

Finally, this project is part of the laboratory and helps to improve its equipment. I hope that some future student can finish the project with this documentation and the solar power plant can be useful for model demonstration.