Designing the solar collector position with the Arduino electronic platform

Peter Vician1,* , Peter Ŏurčanský1, and Jozef Jandačka1
1University of Žilina, Department of Power Engineering, 010 26 Žilina

Abstract. The article deals with the utilizing of solar energy by the means of concentric trough collector based on own design. It focuses on achieving optimal position of the trough with respect to incident radiation during its operation. Due to the collector virtual sun tracking system, which was proved to be ineffective and could not achieve the desired results, the optimum position had to be maintained by manual control of rotation. The measurement of the angle of rotation of the concentric trough is realized using the ARDUINO electronic platform, for which the program was created in the software ARDUINO (IDE) 1.8.4.

1 Introduction

The thesis deals with the use of a concentric trough collector located on the building of University of Žilina. The design of the collector was made according to its own design. Due to its later modification and future application it is necessary to test the collector and determine its performance parameters. The collector has an automatic virtual tracking system, which uses mathematical algorithm to determine the position of the sun and subsequent rotation of the collector. Early measurement has shown some performance deviations due to incorrect trough position. Further tests have shown that the actual tracking system is unable to maintain an optimal position throughout the day. For the purposes of the following measurements, functional monitoring should be ensured. Tilt of the trough can be set manually using the control panel located in the laboratory. The optimal values of tilt angle for time interval of specific day was determined according to calculation of solar angles. However, for the manual operation of the collector and the angle of rotation, it is necessary to provide a response that can be recorded in the form of data. This task is accomplished by the Arduino Mega 2560 Rev3 electronic programmable platform with an additional module MPU-6050 filling function of accelerometer and gyroscope. The program for calculating the angle of rotation and recording the measured values into Excel was created in the freely available ARDUINO (IDE) software 1.8.4. The device itself and the attached modules are powered via the USB A/B connecting data cable. Thus, the device can be programmed and simultaneously powered by one cable. Once the device is connected to an external power source, we can also use it in difficult-to-access areas such as the roof of a building and its operation is limited by the capacity of the battery.

* Corresponding author: peter.vician@fstroj.uniza.sk

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).
However, in our case, for measurement purposes, it is not enough to simply put the measured data on the microSD card but also to get a real-time response. For this reason, the created program displays the measured values in the specified time interval via the serial monitor.

![USB Data Extension 1.1 (LAN) cable and RJ-45 network cable.](image1)

For real-time data transmission, you must ensure that the device is connected continuously to a computer using a data cable. The USB data cable can fully transfer the data to a distance of approximately 10 m, but this is not enough in our case. For data transmission over longer distances the signal has to be amplified by repeater. A better solution is to use an easy-to-get network cable with RJ-45 and a USB/LAN extension to a distance of 30 m. A short measurement of the collector power parameters was performed to verify the device's functionality under realistic conditions. Measured data using the MPU-6050 chip was compared with data obtained from the control panel.

2 Solar tracking systems

The tracking mechanism must be reliable and capable of monitoring the sun with a certain degree of accuracy, even during alternating cloud, which depends on the collector acceptance angle. Another important task that the mechanism provides is to return the collector to its original position at the end of the day or at night. Some systems, in addition to tracking, also perform a protective function and turn the collector out of focus to protect it from hazardous environmental and working conditions such as severe wind shocks, overheating, and failure of the heat flow mechanism. There are different forms of tracking mechanisms from complex to very simple. They can be divided into two broad categories: mechanical and electronic systems. Electronic systems generally have greater reliability and tracking accuracy. These can be further subdivided into mechanisms using electronically controlled motors through sensors that measure the size of solar illumination and mechanisms using computer controlled engines with feedback provided by solar-flow sensors on the receiver [1, 2, 3].

![Parabolic collector with unidirectional positioning.](image2)
Another form of tracking system is called "virtual" tracking, where a traditional sun monitoring unit with sensors that detect the position of the sun has been replaced by a system based on calculating the position of the sun using a mathematical algorithm. The unit is implemented with a 13-bit optical angle encoder mechanically coupled to the collector rotation axis. By comparing the position of the sun and the collector axis with the electronic devices, a signal will be sent to the propulsion system to move to the optimal position [4].

3 Design of measurement device

Arduino is a freely available electronic platform based on easy-to-use hardware and software. Arduino boards can read input values using additional modules. The input can be light on the sensor, finger on the button, Twitter message or other. Any input parameter can be changed to output, e.g. engine activation, LED on, online publishing. The function and usage depend on the set of instructions that are sent to the microcontroller on the board. The Arduino programming language, based on the Wiring programming framework, allows you to write multi-layer software to control devices connected to a wide range of microprocessor boards to create all kinds of creative codes, interactive objects and environments. The Arduino platform is used by a wide community of students, musicians, artists and professionals. It has been used for thousands of projects ranging from everyday objects to complex scientific instruments.

There are several Arduino platforms and a number of sensor types. For our project, the Arduino Mega 2560 Rev3 microcontroller model, the MPU-6050 with accelerometer and gyro function, and the Arduino SD card module were used.

The sensor InvenSense MPU-6050 contains a MEMS accelerometer and a MEMS gyro in one chip. It is very accurate because it contains 16-bit analog digital converter hardware for each channel. The chip captures values in coordinate axes x, y and z simultaneously. The sensor uses an I2C bus to connect with Arduino.

The design of the program was based on a manual for the Arduino programming language and tutorials, which are freely available on the Internet. Program created in ARDUINO (IDE) software 1.8.4 can be theoretically divided into four functional units.

For proper communication of the microcontroller with peripheral devices, it is necessary to address the corresponding pins via bus. This step can be done manually by using physical addresses or automatically by placing the appropriate library in the program.

In the first part of the program we will include the necessary libraries, whose commands we will use. Define the bit length and variable type with the appropriate name. Next, we create a file object to record data to the SD card and define the pin chipset for the SD card module.
In the second part we start one of the main parts of the program, which contains commands executed only once, always when the program is turned on. Commands include initiating communication with the COM serial port and defining the pin of the chipset as the output. Next, initialization of the SD card with the corresponding response in the form of a message on the serial monitor follows. The following is the creation of a text file named Data.txt, in which the beginning of the new measurement is written, and it is simultaneously printed on the serial monitor. If a file is not created, the process failure information is listed. In the last step, we begin communication with the gyro module and read the values recorded by the gyroscope thousands of times every 3 microseconds into the created variables. We divide the recorded values by thousands to get the average value. The last set of commands is important for the next program run, such as the start of integrated time logging, resetting the calculation loop, and the write loop. Since, in our case, no real-time module is used to fill the required I2C pin with the gyro module, time will be measured using an integrated timer that will count down seconds after the measurement starts. Since the number of calculations per time unit is different from the number of entries, different timers have to be loaded.

The third part contains a calculation loop that reads the gyro value and subtracts it from the previous value, calculating the difference. This value is converted to the climb angle. Since the sinus function works with radians, the obtained values have to be converted. The values obtained with the accelerometer are processed similarly and the resulting tilt values are obtained by the sum of the component of the gyroscope and the accelerometer at the given ratios. To avoid excessive variations in output values, damping has been introduced. This means that the output values will be only 10 % of the new value and 90 % of the previous ones. The tilt angle values and the measurement time are displayed on the serial monitor at a specified interval of 2 seconds, with the calculation loop running every 4 microseconds.

The last part of the program is used as a setup and calibration function of the MPU-6050. By sending the specified values to the individual microchip registers, we can adjust the sensitivity of the gyroscope and accelerometer by changing the measuring range.

4 Results comparison

The created program was loaded on the measuring device placed on the counterweight frame of the collector. This part is a solid plane intersecting the axis of the concentric trough and is the ideal place for mounting the measuring device. Since the solar collector is located in an outdoor environment where it is exposed to environmental influences such as rain, snow, wind, sun, or birds, it is essential to protect the surface connections of the measuring device.

Fig. 4. Placement of the measuring device with protective case.
In the second part we start one of the main parts of the program, which contains commands executed only once, always when the program is turned on. Commands include initiating communication with the COM serial port and defining the pin of the chipset as the output. Next, initialization of the SD card with the corresponding response in the form of a message on the serial monitor follows. The following is the creation of a text file named Data.txt, in which the beginning of the new measurement is written, and it is simultaneously printed on the serial monitor. If a file is not created, the process failure information is listed. In the last step, we begin communication with the gyroscope module and read the values recorded by the gyroscope thousands of times every 3 microseconds into the created variables. We divide the recorded values by thousands to get the average value.

The last set of commands is important for the next program run, such as the start of integrated time logging, resetting the calculation loop, and the write loop. Since, in our case, no real-time module is used to fill the required I2C pin with the gyro module, time will be measured using an integrated timer that will count down seconds after the measurement starts. Since the number of calculations per time unit is different from the number of entries, different timers have to be loaded.

The third part contains a calculation loop that reads the gyro value and subtracts it from the previous value, calculating the difference. This value is converted to the climb angle. Since the sinus function works with radians, the obtained values have to be converted. The values obtained with the accelerometer are processed similarly and the resulting tilt values are obtained by the sum of the component of the gyroscope and the accelerometer at the given ratios. To avoid excessive variations in output values, damping has been introduced. This means that the output values will be only 10% of the new value and 90% of the previous ones. The tilt angle values and the measurement time are displayed on the serial monitor at a specified interval of 2 seconds, with the calculation loop running every 4 microseconds.

The last part of the program is used as a setup and calibration function of the MPU-6050. By sending the specified values to the individual microchip registers, we can adjust the sensitivity of the gyroscope and accelerometer by changing the measuring range.

### Results comparison

The created program was loaded on the measuring device placed on the counterweight frame of the collector. This part is a solid plane intersecting the axis of the concentric trough and is the ideal place for mounting the measuring device. Since the solar collector is located in an outdoor environment where it is exposed to environmental influences such as rain, snow, wind, sun, or birds, it is essential to protect the surface connections of the measuring device.

![Fig. 4. Placement of the measuring device with protective case.](image)

For proper fitting, protection, and ease of access when replacing chips or removing a microSD card, the device was stored in a modified plastic container with a removable cap (Figure 4). Device communication and initialization is controlled via a combination of USB/LAN cables, as mentioned earlier, through the shaft to the laboratory premises.

![Fig. 5. Measured tilt angle according control panel and designed device.](image)

In order to verify the correct operation of the device, the tilt angle indicated by control panel display and the angle obtained from the designed device located on the collector was compared (Figure 5). The graph was designed for reference angles from -90° to 90° in 10° increments. Based on the geographic position and the location of the collector, it was found that the 180° range is sufficient to provide optimal position throughout the day. Angle -90° is the standby position (trough facing downwards) and the 90° is a fully open collector with the focus in the vertical axis. On the chart we can see that the values measured by Arduino in the range of -90° to 0° are higher than the values obtained from the control panel and vice versa in the range of 0° to 90°.

![Fig. 6. The values of deviation of the measured tilt angle using the Arduino device.](image)

The measurement results show relatively small deviations in the area around the horizontal position of the gyroscope, maximum of 2.53° at a 30° inclination. Deviation values gradually increase with increasing tilt angle to the vertical position. The highest deviation of 11.62° was measured in a vertical position of 90°. The measurement shows that the deviances range from 0.15° to 11.62°.
Taking into account the effect of the deviation from the optimal position on the concentric solar collector power, the observed deviations represent a significant decrease in collector efficiency. For this reason, the device is not suitable for precise control of the collector operation. The detected deviations may be caused by the incorrect formulation of the microchip program MPU-6050 and must be subjected to a closer analysis.

This article has been released with financial support from the European Union and project support. We support research activities in Slovakia. The project is co-financed by EU funds. Project title: "Research on new methods of heat conversion from RES to electricity using new progressive thermal cycles" ITMS 26220220117 and APVV-15-0790 "Optimization of biomass burning with low melting point of ash".

References

1. S.A. Kalogirou, Solar energy Engineering (Academic Press Elsevier, California, 2014)
2. S.A. Kalogirou, Solar Energy 57 (1996)
3. F. Briggs, Tracking-Refinement Modeling for Solar-Collector Control. (Sandia National Laboratory, Albuquerque) (1980)
4. K.D. Boultinghouse, Development of a Solar-Flux Tracker for Parabolic-Trough Collectors (Sandia National Laboratory, Albuquerque, 1982)
5. G. Brakmannet, Concentrated solar thermal power - now! (ESTIA, Brussels, 2005)
6. G.L. Morrison, Solar collectors (James and James, London, 2001)
7. K. Chia-Wei, Y. Pei-Shan, Ch. Wen-Chey, Ch. Keh-Chin, Procedia Engineering, 79 (2014)
8. H. Caiqin, X., Procedia Engineering 121 (2015)
9. A. Rabl, Solar Energy 18 (1976)
10. H. M. Guven, R. B. Bannerot, Solar Energy 36 (1986)
11. A. Čaja, P. Nemec, M. Malcho, AIP Conference Proceedings, 1608 (2014)
12. P. Nemec, M. Smitka, M. Malcho, Scientific world journal, 2014 (2014)