Angular Position Sensor for Angular Position Tracking

Gunawardana S.A.
Department of Electrical & Electronic Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e18125@eng.pdn.ac.lk

Herath W.M.V.S.
Department of Electrical & Electronic Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e18134@eng.pdn.ac.lk

Weerasingha W.K.H.M.
Department of Electrical & Electronic Engineering
University of Peradeniya
Peradeniya, Sri Lanka
e18383@eng.pdn.ac.lk

***Abstract*—This paper describes an angular position sensor that is used for angular position tracking in an antenna propagation testing setup. Initially, this paper describes briefly introduces the sensor and the intended application. Then, this paper describes the principle of operations along with the particular circuit diagrams. Also, it describes the calibration, verification, testing, and validation setups. Lately, this paper has targeted the observational results and the final specifications of the sensor. Finally, describes the weaknesses, strengths, and overall comparison of the sensor.**

I. Introduction

Angular position sensors are a kind of measuring equipment that is used to track down the angular position of a rotating object. Those angular position trackers are frequently used on vast applications such as navigation systems, radar systems, etc. There are many types of angular position sensors with various sensing methods that can be seen in the modern market. In this project, track the angular position of a rotating antenna connected to a stepper motor and a turntable to find out the better signal receiving area to the antenna. In the practical scenario, there will be two antennas in this setup. One of them is receiving antenna while the other antenna is an emitting antenna. The standard radiation measurement system procedure has shown in figure 1 [14]. As for the project application, there is an optical encoder to find out the RMS error which obtains from the stepper motor that has been used to rotate the turning table. So, the sensor that, developed here can be used to recheck the angle, which is given by the stepper motor, into one-degree accuracy. Here, assume that the angle given by the optical encoder is accurate more than the stepper motor angle because many possible causes can happen with the stepper motor such as backlashes, friction forces, and heating issues that leads the stepper motor angle to a more inaccurate value.

The optical encoder sensor is designed with a gray code, a stepper motor, and an Arduino-based light sensor [1]. As for the gray code is designed a 9-bit circular gray code with 1° (degrees) least count on it [2], [3]. With that, the light sensor is equipped with one-degree accuracy, and it can be further improved for a 0.5-degree accuracy along with applicational purposes. The light sensor has been operated with an Arduino Uno board and programmed to control the light sensor that was designed. In the light sensor, we used photo resisters and some ICs to take the readings as a gray value and convert it to a decimal value, and then finally, display the angular position on the LCD display. With that, the user can easily give an arbitrary angle to rotate for the stepper motor and check whether it is correct or incorrect using the Output angle displayed on the LCD panel.

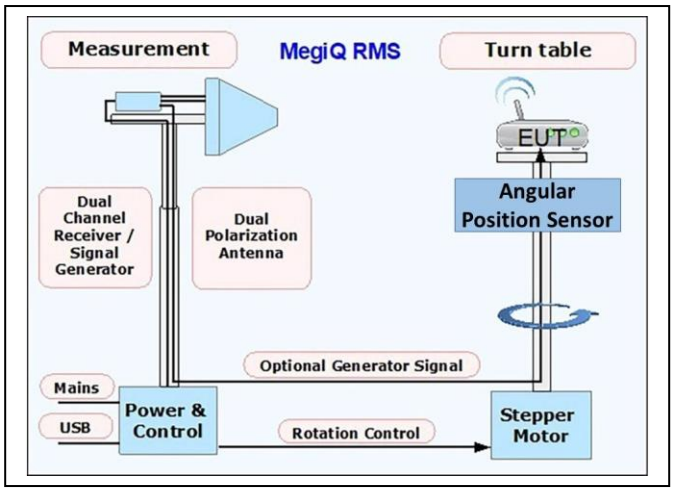


Fig. 1: Radiation Measurement System

II. Initial Specifications

- The light sensor can sense a full rotation with a one-degree step size.
- A step motor with a torque about 3 Nm, will be used here to rotate an antenna that weighs around 2 kilograms.
- Increment per 200 step motor = 1.8°
- Advance per 1 revolution on input = 360'
- Gear ratio 1:1 latency
- Travel 0-360°+
- Rotary table diameter 6 inch
- Device; length = 12-inch, width = 8-inch, height = 6 inch

- Device weight 2.5 kg
- Maximum input torque 50 oz-in

III. METHODS

A. PRINCIPLE OF OPERATION

The optical rotary encoder sensor is consisting of 9 LEDs, 9 photoresistors, two types of logic ICs, an LCD display, resistors, a 9bit rotary grey code scale, and an Arduino Uno board to operate the sensor. When the setup is powered up, all LEDs going to be lit up and the grey encoder has to be rotated along with the rotating shaft. Then the light going through the encoder reaches the photoresistors and those photoresistors can sense the angle with the intensity pattern through the encoder. Figure 2 shows the schematic diagram of the light sensor. Those photo-resistor outputs are in a grey code format, and they have to be converted to a decimal value before displaying in the LCD panel. So, the LDR outputs are sent to the ICs to convert their format to a binary value and these binary values are converted again to decimal values with an Arduino program and finally, the measured angle is displayed on the LCD panel. In order to do these calculations and the measurements, many components have been placed inside the circuit with a unique task assigned for them. Figure 6 shows the circuit diagram of the sensor which has been designed using proteus. [11]

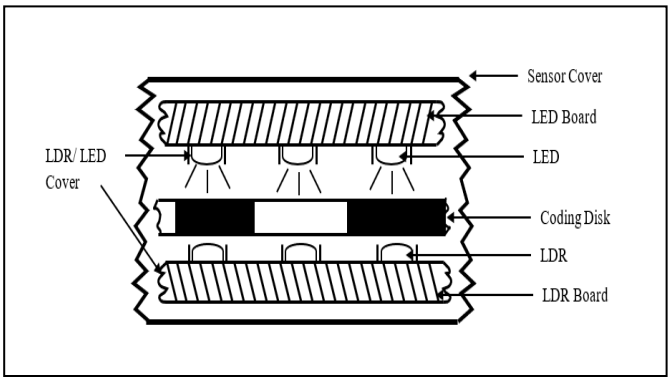


Fig. 2: Structural schematic for the sensor

To bring about straightforward data encryption and decryption, ICs have been used in the light sensor. Two types of logic ICs have been used in the sensor, they are 74HC00 and 74HC86. These ICs are made of advanced silicon-gate CMOS technology which means that they are complementary metal-oxide semiconductors. Therefore, these ICs can be operated in a wide range of voltages. Also, they have a very high noise immunity with them. Here, both ICs are from the high-speed CMOS series. These ICs are highly sensitive to static charges. Hence, all non-used pins must be grounded to get the accurate output.

74HC00 is a Quadruple 2 input NAND gate and totally three 74HC00 ICs have been used in the sensor to output the grey code value [4]. 74HC00 IC is consists of two input four NAND logic gates. Pinout diagram of the 74HC00 IC is shown in figure 3. The other type of IC used in the sensor was 74HC86 and it is a Quadruple 2 input XOR gate [5]. It consists of four two-input XOR gate logics. Pinout diagram of the 74HC86 IC is shown

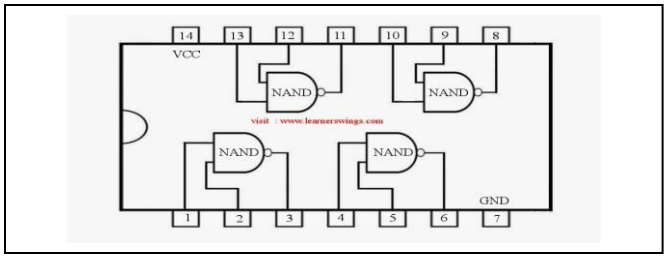


Fig. 3: Pinout diagram of 74HC00 logic IC

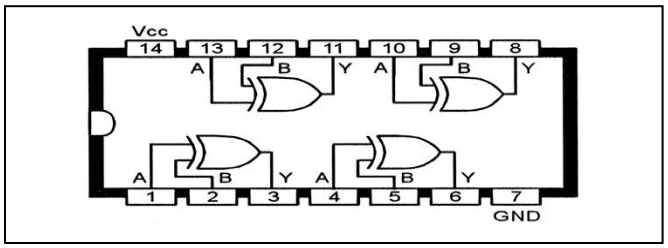


Fig. 4: Pinout diagram of 74HC86 logic IC

in figure 4. This IC has been used here to convert the grey value to a binary value. For this sensor, the supply voltage to each IC has been kept as 5V because all these ICs can be operated in a voltage between 2.0V from 6.0V.

Resistors that have been used in the circuit are defined according to the E24 series resistors and all these resistors have a 5% tolerance value. Since the tolerance percentage is low it helps to protect the components used in the sensor.

Nine PGM5616D cadmium sulfide LDRs which are 5mm in diameter have been used in the circuit to make sure they are suitable for the size of the grey code [6]. The ambient temperature range of these LDRs is -30C to +70C. This wide temperature range would be handy because the sensor is operated mostly in outdoor conditions.

Figure 5 shows that these cadmium sulfide LDRs have a peak spectral response at about 500nm wavelengths and the LDR is more sensitive near this range [9]. Therefore, using green LEDs will be the best color because LDRs that have been used here have a spectral peak of 560 nm. Since green LDRs have a wavelength of about 550nm, they will work efficiently with the LDRs in the sensor. Considering all these reasons, green color LEDs are used in this project.

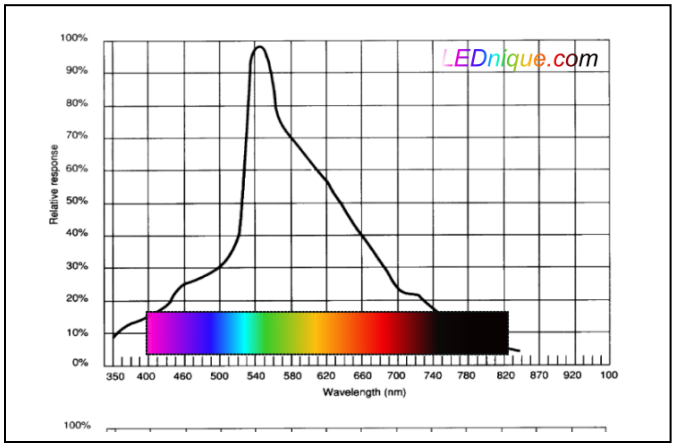


Fig. 5: LDR spectral response with visible light spectrum superimposed

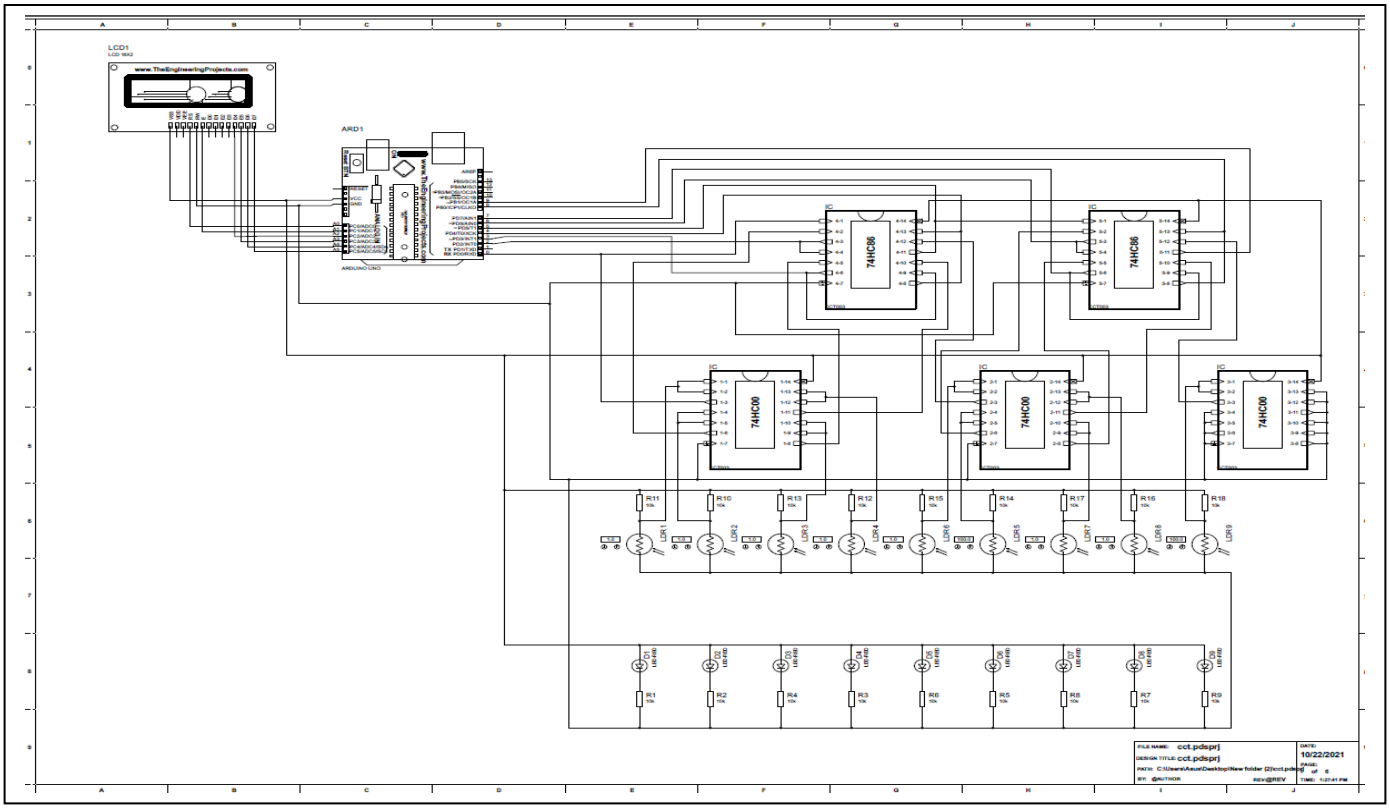


Fig. 6: Circuit diagram of the sensor

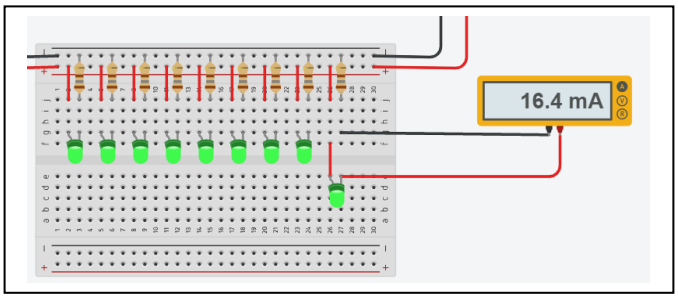


Fig. 7: Measured value of current through LED

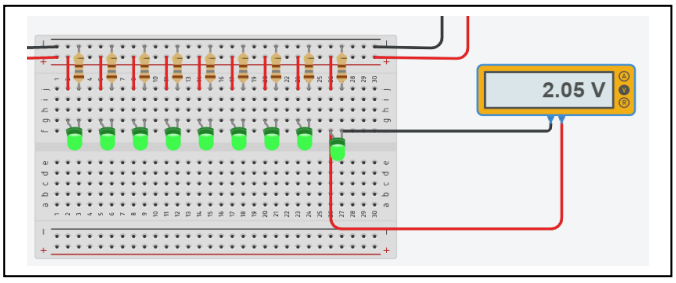


Fig.8: The measured value of voltage across LED

Here, nine MCL053GD Green color LEDs have been used with 5mm width to compact the size of the sensor [7]. The viewing angle of these LEDs is 45 degrees and to get better performance all LEDs have to be covered separately. Otherwise, it will affect the performance of the other LDRs, and the output will become erroneous. To protect the LEDs from high voltages, 180Ω resistors have been used in the circuit.

Figure 7 shows the current through each resistor has been measured in the simulation and it is 16.4 mA. According to the datasheet, the recommended operating current value range is 12mA to 20 mA.

Figure 8 shows the voltage across the LED In the simulation the voltage across each LED is 2.05V and that can be accepted because, these LEDs require a forward voltage in between 1.7V to 2.6V for the better performance. Also, the 10% tolerance of the resistors would not be affected harmfully to the LEDs. Both these simulations have been done using tinkercad.[12]

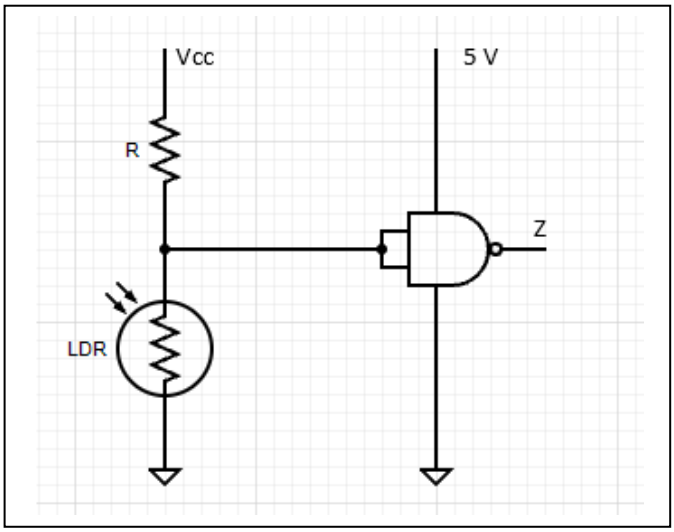


Fig.9: Voltage divider circuit

TABLE 1: NAND GATE TRUTH TABLE FOR VOLTAGE DIVIDER CIRCUIT

INPUT	OUTPUT
0	1
1	0

In the sensor circuit diagram, there used nine voltage divider circuits as shown in figure 9. When the feeding voltage to the NAND gate is greater than half of the supplied voltage to the NAND gate, the output from the gate will be digital “1”. Table 1 shows the truth table for the NAND gate which is used in the voltage divider circuit. For, that particular outcome to take place, the resistance of the resistor (R) must be lower than the LDR resistance. When the light intensity increases the resistance of the photoresistor decreases. Hence, when the R resistor value is low, the LDR must expose to more light intensity to give the desired output. Since that, for the practical scenario lower R resistor value is much suitable. Here for the simulation process, a 1 kΩ resistor has been used to give a better output. But light from the other LEDs can affect the light intensity on other photoresistors. So, the most practical R resistor value must be defined after implementing the sensor physically.

```
//Sensor project
float angle;
// import libraary for LCD
#include<LiquidCrystal.h>
LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);

void setup()
{
  Serial.begin(9600);
  lcd.begin(16, 2);
}

void loop()
{
  // Read input values from LDRs
  int bval0 = digitalRead(0);
  int bval1 = digitalRead(2);
  int bval2 = digitalRead(3);
  int bval3 = digitalRead(4);
  int bval4 = digitalRead(5);
  int bval5 = digitalRead(6);
  int bval6 = digitalRead(7);
  int bval7 = digitalRead(8);
  int bval8 = digitalRead(9);

  // get decimal value
  angle = (bval0*(256)) + (bval1*(128)) + (bval2*(64)) + (bval3*(32)) +
  (bval4*(16)) + (bval5*(8)) + (bval6*(4)) + (bval7*(2)) + (bval8);

  Serial.println(angle);

  lcd.setCursor(0,0);
  lcd.print("Angle = ");
  lcd.print(angle);
}
```

Fig. 10: Arduino code for the sensor

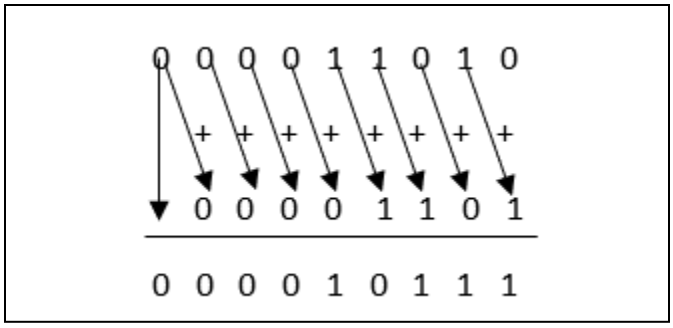


Fig. 11: Converting a binary value to gray code value

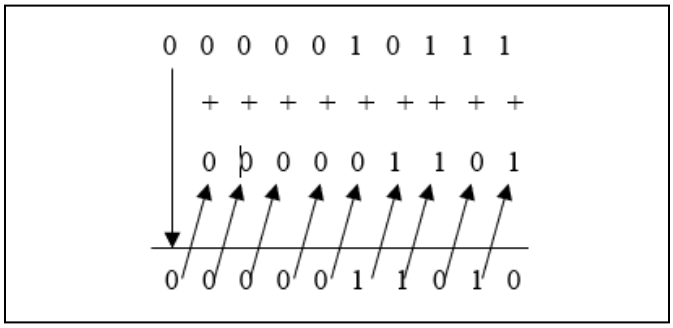


Fig. 12: Converting a gray code value to binary value

TABLE 2: TRUTH TABLE FOR XOR

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

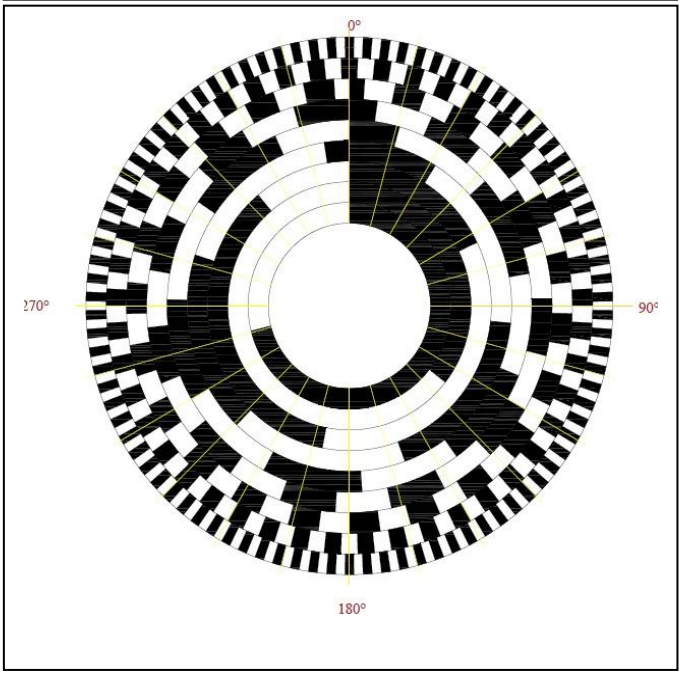


Fig.13: Nine-bit grey code design

Grey values have been used instead of binary values because in grey code the transition between two adjacent values only changes by a single bit. So, errors can be reduced with a grey code encoder than a binary encoder.

The grey code has been designed with AutoCAD and it consists of nine circular strips which represent the nine bits of the grey value[10]. Figure 13 illustrates the cad design of grey code. The most significant bit has been placed near the origin of the circle while the least significant bit has been placed near the circumference of the circle. The grey code is rigidly fixed to the rotating shaft, and that is the only part rotating along with the rotating shaft. The LEDs have been placed above the grey code encoder and the LDRs have been placed below the grey code stationary without rotating along with the rotating shaft. Since the grey code is an important part of the sensor it has to be printed on a rigid sheet without any deformations with natural environmental causes such as humidity and temperature. A figure which describes the placement of the grey code, LDRs, and LEDs has been figured above in figure 2.

Converting grey values to a binary value can be used to sign an XOR logic operation. Table 2 shows the truth table for the XOR logic gate. Therefore, in this project, 74HC86 logic ICs have been used to convert 9-bit grey code values which are the outputs of 74HC00 ICs to a 9-bit binary value. In Figures 11 and 12, conversion between binary to the grey of the decimal value 26 has been shown. An Arduino program has been used here to convert the binary output to a decimal output and transfer it to the LCD panel to display the output. Figure 10 shows the Arduino code which has been used for the above purpose.

An LCD has been used in the sensor to display the output angle in real-time. Here, a “waveShare LCD1602” display has been used and the operating voltage of this display is between -0.3V to +7.0V [8].

To do all the pre-mentioned tasks, a simulation has been designed using tinkercad to simulate and preview the final design of the sensor. Working simulation has been recorded and the link for the simulation is given here.
Link:
https://drive.google.com/drive/folders/1VpnGmxLCICW-BQI8CSrBgfL0Se0GyrRS

B. TESTING METHOD

One complete rotation with checking all angles from 0 degrees to 360 degrees, can be given as a testing setup. With that, any malfunctioning issues of the sensor can be identified. This testing process can be done by manually rotating one complete revolution.

C. CALIBRATION METHOD

Completing one full revolution starting from zero degrees, with a step size of 15 degrees can be used as a calibration method because while rotating with 15-degree step size, all LDRs will achieve logic 1 and 0 states at least once. With that calibration, the user can identify whether the sensor is working perfectly, or it has some issues with the electronic components inside the sensor. ICs, photoresistors, and all other electronic components could have some issues with time. But all issues can be identified with the calibration setup and necessary steps such as replacing the components can be done to eliminate those issues. For done this calibration part perfectly, the rotating table has been marked with 15-degree partitions and those predefined angles are given in figure 14.

Digit	Binary	Gray code
0	000000000	000000000
15	000001111	000001000
30	000011110	000010001
45	000101101	000111011
60	000111100	000100010
75	001001011	001101110
90	001011010	001110111
105	001101001	001011101
120	001111000	001000100
135	010000111	011000100
150	010010110	011011101
165	010100101	011110111
180	010110100	011101110
195	011000011	010100010
210	011010010	010111011
225	011100001	010010001
240	011110000	010001000
255	011111111	010000000
270	100001110	110001001
285	100011101	110010011
300	100101100	110111010
315	100111011	110100110
330	101001010	111101111
345	101011001	111110101

Fig. 14: Predefined angles for the calibration setup

If the rotating table is tilted aside, it may affect the output values directly. So, to overcome this issue a water level has been installed on the surface to assure that the rotating table is horizontal. Also, the user can observe the grey code before the procedure to check if there are any defects with the grey code.

D. VERIFICATION METHOD

To verify that the sensor is working as expected, different angle measurements have to be done in different environmental conditions.

Since the sensor may be vulnerable to light conditions and temperature conditions same angle set has to be measured in different environmental conditions such as low/high light conditions and low/high temperature conditions. After verifying the sensor is working perfectly in normal environmental conditions, the user can guarantee that the measurements are in the expected accuracy without any deviations from the actual measurements.

E. VALIDATION METHOD

Since the final objective of this project is to measure the angle of an antenna that is driven by a stepper motor, a set of measurements have to be done after connecting the stepper motor to the setup. With that, we can assure that this sensor can be used for the given project. There might be some latency issues after connecting the step motor. So, those issues have to

be checked whether that they are acceptable to ensure an accurate output as expected. Also, there might be some rubbing actions between components while the stepper motor and the main shaft are rotating and that issue has to be reduced to guarantee the accuracy of the sensor.

IV. Results

When setup is connected to a power supply, it begins to display the actual angle on the LCD panel and the user can define an angle for the stepper motor to rotate. With that, the stepper motor angle can be identified as an input and the angle given from the sensor can be identified as the output. So, an input-output characterization graph can be drawn from the data which have been obtained by each measurement. The x-axis and the y-axis can be denoted by the input stepper motor angle and the output angle from the sensor respectively. In the ideal situation, when the stepper motor angle is the same as the actual angle, a y=mx graph can be expected, but in the practical observations, there might be few deviations from the ideal characteristics graph.

The results from the calibration setup can be calculated and recorded for future purposes and from that RMS value of the error, any angle obtained from the stepper motor can be corrected to an accurate angle.

$$RMS = \sqrt{\frac{1}{n}\sum_i (x - x_i)^2} \quad (1)$$

Here, RMS is the root mean square values of errors while n is the number of measurements, x is the input angle to the stepper motor and x_i is the angle measured by the sensor.

Since the RMS value of the error from the stepper motor has to be calculated, the stepper motor reading and the actual angle reading have to be taken down. For that, the stepper motor has to be powered up and along with that the shaft and the rotary encoder have to be rotated while the light sensor stays stationary and measures the actual angle. The output of the nine reading heads from the photoresistors are transferred to the light sensor and those outputs are converted to a decimal value and stored in real-time for a complete one revolution. With that, the given stepper motor angle and the error that has been happened with each angle can be calculated and those two parameters can be illustrated in a plot to finalize the RMS error value of the stepper motor. The error value in degrees can be represented by the Y-axis and the stepper motor angle in degrees can be represented by X-axis. After plotting this for one complete revolution, a graph like shown in figure 15 can be expected.

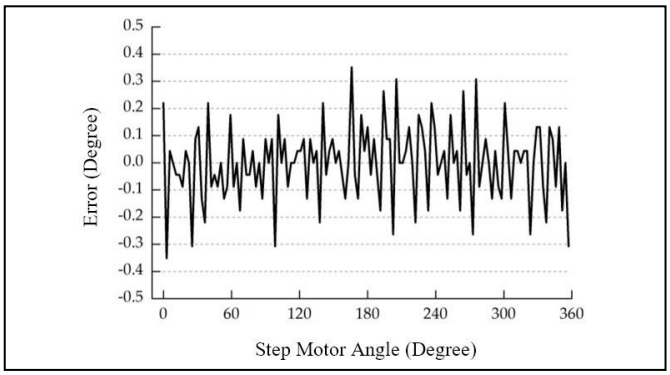


Fig. 15: Error vs stepper motor angle

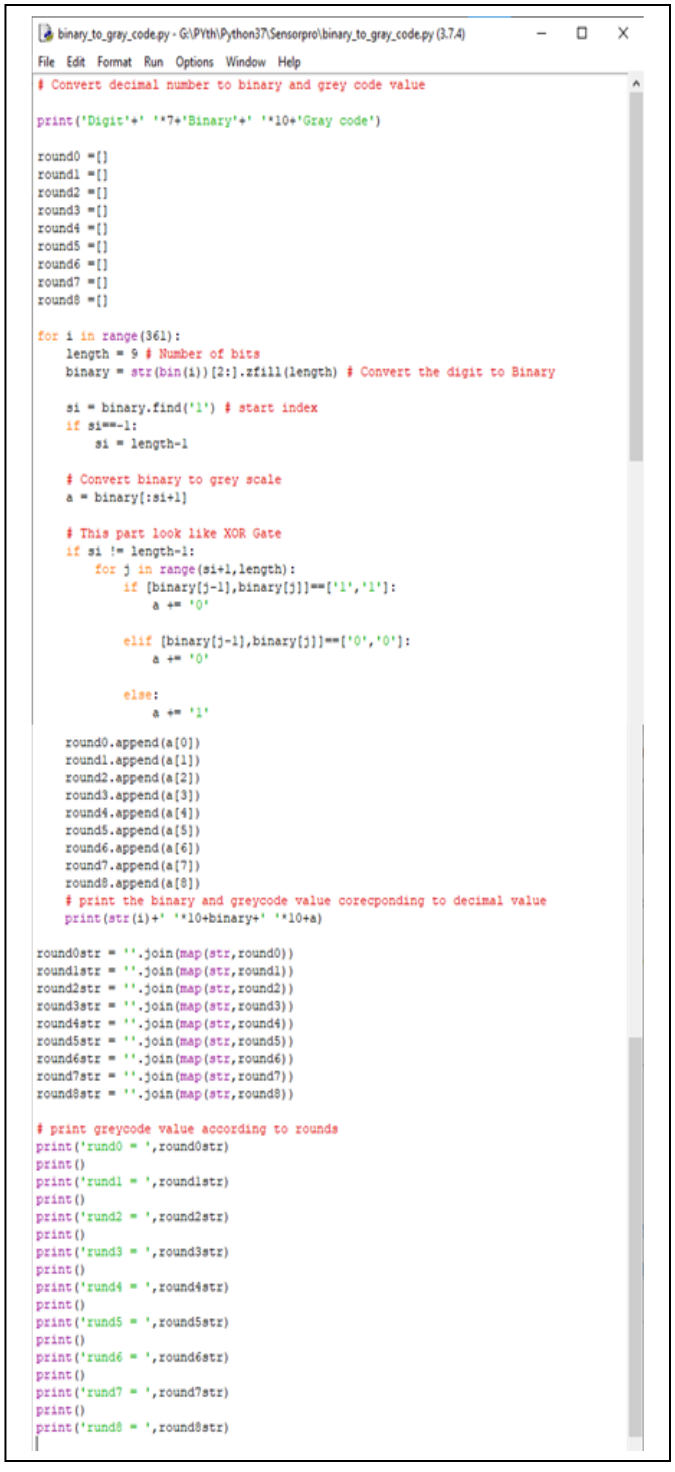
V. Discussion

We faced some main difficulties while designing the sensor. Major problem was that we were unable to brainstorm physically due to the pandemic situation. Also due to that, we were unable to find the electronic equipment that was needed to physically implement the sensor as expected at the beginning of the project. Therefore, the statistical analysis part and the input-output characteristic diagram could not be done properly, but we discussed that and came up with some ideas to do those analysis parts after physically implementing the sensor in the future. The gray code that was designed has to be printed on a special film roll, but without the lab facilities, it is impossible to make it happen. But we asked our resource person and he told us it can be printed in the university lab just after the university reopens. Hence, we were unable to implement our sensor without the grey code, but we designed our light sensor and simulated it on "Tinkercad" to get a better idea about the sensor and here we attached some figures from the working simulation.

There was quite a doubt about the least value that can be measured with the sensor, then the least value has been marked as one degree to maintain minimum errors in the sensor. The initially designed sensor consists of nine analog pins, and an Arduino Mega board was needed. In this design, the sensor can be simply developed with an Arduino program instead of using any electrical components. But, if an Arduino mega board has been used here, some extra pins will remain, and the cost of the sensor also will rise. After using ICs in the sensor, digital pins can be used instead of analog pins and a display can be also connected to the sensor to display the output. Also, with this design users can use an Arduino Uno, Mega, or a Nano board to operate the sensor and that is a huge benefit to the user Also, ICs are a bit quicker than using an Arduino program to do the calculation part. So, the latency errors have been minimized after using ICs in the sensor.

Using an absolute encoder is suitable for this project than using an incremental encoder because an absolute encoder has some features that would be handy in this project. In an absolute encoder, the position angle is not lost if the power supply is lost, and continuous reading of position is not required since each degree has its own grey value to determine its position.

To obtain grey values, a python program has been created that is shown in figure 16, and it simply displays all grey values within the given range [15]. So, with the help of that python program, a 9-bit encoder has been created to use with the light sensor. Here, all binary and grey values are consisting of 9 bits because to illustrate 360 distinct partitions on the encoder it needs a minimum 9-bit binary value.



```
binary_to_gray_code.py - G:\PYth\Python37\Sensorpro\binary_to_gray_code.py (3.7.4)
File  Edit  Format  Run  Options  Window  Help
# Convert decimal number to binary and grey code value

print('Digit'+' '*7+'Binary'+' '*10+'Gray code')

round0 =[]
round1 =[]
round2 =[]
round3 =[]
round4 =[]
round5 =[]
round6 =[]
round7 =[]
round8 =[]

for i in range(361):
    length = 9 # Number of bits
    binary = str(bin(i))[2:].zfill(length) # Convert the digit to Binary

    si = binary.find('1') # start index
    if si==-1:
        si = length-1

    # Convert binary to grey scale
    a = binary[:si+1]

    # This part look like XOR Gate
    if si != length-1:
        for j in range(si+1,length):
            if [binary[j-1],binary[j]]==['1','1']:
                a += '0'

            elif [binary[j-1],binary[j]]==['0','0']:
                a += '0'

            else:
                a += '1'

    round0.append(a[0])
    round1.append(a[1])
    round2.append(a[2])
    round3.append(a[3])
    round4.append(a[4])
    round5.append(a[5])
    round6.append(a[6])
    round7.append(a[7])
    round8.append(a[8])
    # print the binary and greycode value corecponding to decimal value
    print(str(i)+' '*10+binary+' '*10+a)

round0str = ''.join(map(str,round0))
round1str = ''.join(map(str,round1))
round2str = ''.join(map(str,round2))
round3str = ''.join(map(str,round3))
round4str = ''.join(map(str,round4))
round5str = ''.join(map(str,round5))
round6str = ''.join(map(str,round6))
round7str = ''.join(map(str,round7))
round8str = ''.join(map(str,round8))

# print greycode value according to rounds
print('rund0 = ',round0str)
print()
print('rund1 = ',round1str)
print()
print('rund2 = ',round2str)
print()
print('rund3 = ',round3str)
print()
print('rund4 = ',round4str)
print()
print('rund5 = ',round5str)
print()
print('rund6 = ',round6str)
print()
print('rund7 = ',round7str)
print()
print('rund8 = ',round8str)
```

Fig. 16: Python code to convert decimal numbers to grey code values

The main weakness of this sensor can be given as the size of the sensor. In order to compile with the size of the components of the light sensor, the grey code has to be quite large to maintain an accurate measurement. So, the size of the final design has to be slightly increased, but it can be managed after few modifications in the sensor. Also, the initial installation part would be tricky of this sensor because it consists of some moving mechanical parts such as a stepper motor and a turning table. Furthermore, there are some minor weaknesses such as dimming of LEDs over time, grey code can be covered with contaminants such as dirt and oil. High temperatures can be affected to the sensor since there are LEDs and photoresistors in it. Also, high temperatures can affect the dimensions of the grey code.

Since a non-contact encoder has been used here, the reliability of this encoder is inadequate, and also the resolution of the encoder can be further improved with the applicational purposes.

Compared with the commercial optical encoders, this sensor can be declared as a cost-friendly sensor. There are plenty of varieties of optical encoders in the modern market. Normally, they have a simple technology than inductive encoders. “OCD-S401G- 0016-S060-PRQ” is an absolute rotary encoder that was manufactured by POSITAL company [16]. POSITAL is a sensor manufacturer for motion control and safety assurance systems. This is a 16-bit optical rotary encoder. The accuracy of this absolute rotary encoder is ±0.0220° for (14 – 16 bit) and ±0.0439° for (≤13 bit). There is a Gray Scale in this encoder to measure the angular position. The power consumption of this device is less than 1.5W and the start-up time is less than 1s. These features come in handy in commercial industries. In this product, reverse polarity protection and short circuit protection can be seen. As an environmental specification of this product, this product has an IP65 protection class for the shaft and the housing. The ambient temperature range of this sensor is from -40℃ to 85℃. The friction torque is less than 3Ncm at 20 ℃. This is a small product. So, the weight of this product is almost 285g.

VI. Conclusion

From this sensor, we can find out the error of the stepper motor for a given known angle. Since the error could be positive or negative the RMS value for the error has to be defined after taking some practical measurements with the sensor. Here we assume that the light sensor is giving the accurate angle and the stepper motor angle can be defected because of backlashes, friction, and other possible causes. After calculating the RMS value of the error, the stepper motor angle can be corrected and the correct angle can be used in further calculations of the antenna propagation purposes.

VII. References

[1] Y. Sugiyama, Y. Matsui, H. Toyoda, N. Mukozaka, A. Ihori, T. Abe, M. Takade, S. Mizuno, "A 3.2 kHz, 14-Bit Optical Absolute Rotary Encoder with a CMOS Profile Sensor," in IEEE Sensors Journal, vol. 8, no. 8, pp. 1430-1436, Aug. 2008, doi: 10.1109/JSEN.2008.920709[online]. [Accessed on: Aug. 08, 2021]

[2] Fan Zhang, Hengjun Zhu, Kan Bian, Pengcheng Liu and Jianhui Zhang, “Absolute Position Coding Method for Angular Sensor-Single-Track Gray Codes”. *Journal of Sensors (Basel, Switzerland)*, vol. 18, no. 8, pp. 2728, Aug. 08, 2018. Available doi: 10.3390/s18082728[online]. [Accessed on: Sep. 23, 2021]

[3] Das. Jadav Chandra, De. Debashis, “Reversible Binary to Grey and Grey to Binary Code Converter using QCA”. *IETE Journal of Research*, vol.

61, no. 3, pp. 223-229, May 04, 2015. Available doi: 10.1080/03772063.2015.1018845[online]. [Accessed on: Aug.08, 2021]

[4] Diodes incorporated, "QUADRUPLE 2-INPUT NAND GATES," 74HC00 datasheet, January 2013.

[5] Diodes incorporated, "QUADRUPLE 2-INPUT EXCLUSIVE OR GATES," 74HC86 datasheet, January 2013.

[6] Token electronics, "CdS Light-dependent photoresistors", PGM CdS photoresistors datasheet, 2010.

[7] Multicomp Pro, "Round LED, Green, 5mm", MCL053GD datasheet, Sep. 17, 2019.

[8] WaveShare electronic, "WaveShare LCD1602", WaveShare LCD1602 datasheet, Oct. 2020.

[9] lednique.com, "Light dependent resistor (LDR)," *LEDnique*, 2021[online]. Available: http://lednique.com/opto-isolators-2/light-dependent-resistor-ldr/ [Accessed: Aug. 17,2021].

[10] *AUTOCAD,* (2018), Autodesk, [computer software], Available: https://www.autodesk.com

[11] *Proteus 8 Professional*, Labcenter Electronics Ltd, [computer software], Available: https://www.labcenter.com

[12] *Tinkercad (2021),* Autodesk, [web editor]. Available: https://www.tinkercad.com/

[13] *Circuit Diagram (2021),* circuit diagram, [web editor]. Available: https://www.circuit-diagram.org

[14] akoriot.com, "MegiQ radiation measurement systems", 2021[online]. Available: https://www.akoriot.com/radiation-measurement/

[15] *Python 3.7.4*, Python software foundation, Available: https://www.python.org/

[16] posital.com, "IXARC Absolute Rotary Encoder ", Sep. 4, 2019 [Online]. Available: https://www.posital.com/en/products/absolute-encoders/absolute-encoder-finder/OCD-S401G-0016-S060-PRQ/123916201/detail.php [accessed: Sep. 29,2021]