To: Your ta, Mechatronics TA
From: Gary Oak, Mechatronics Student (be sure to initial or sign here)
Date: August 30, 2001 (replace this with the date of submission)
Subject: Encoder design lab report

Introduction: The purpose of this memo is to present and discuss the results from the encoder design experiment. In this exercise, a 4-bit gray code absolute encoder was designed, constructed, and analyzed. Encoders are useful in providing sensory feedback for position in rotary systems. In analyzing the gray code encoder, important issues such as resolution, binary math, and logical truth tables were addressed.

Methods and Procedures: In this exercise, a 4-bit gray code absolute encoder was constructed with black foam board. An absolute encoder is a disk that is divided into tracks that radiate outward from the center and that is divided into equal sections through the center. Each track provides one bit of information, and the number of sections corresponds to the resolution of the encoder. Encoder resolution is a measure of the smallest angle or displacement that the sensor can discern. The formula that relates the number of bits to the resolution of the encoder is as follows:

$$res = \frac{360°}{2^n} \qquad (1)$$

where *res* is the resolution of the encoder in degrees, and *n* is the number of bits. The number of divisions on the encoder was 2^n, or 16 divisions, with 4 tracks corresponding to the 4 bits of desired information. When something is said to have a high resolution, that means that the number of bits is relatively high, which corresponds to a small actual resolution, *res*.

The absolute encoder is a device that reads position in a digital fashion. In other words, the absolute encoder provides information in the form of a binary number, which must be converted to a decimal number to make sense to people. The conversion from binary to decimal number systems goes as follows:

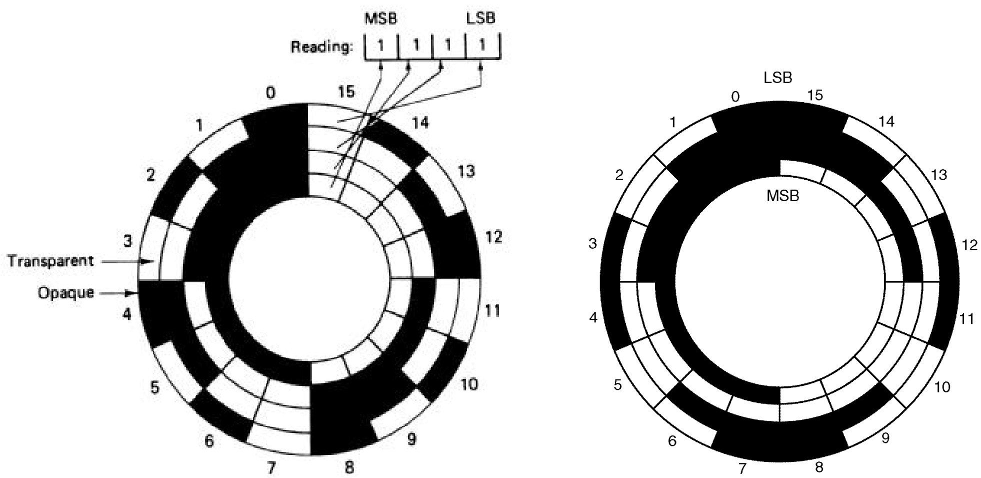
$$n_d = 2^{n-1} b_{n-1} + \cdots + 2^2 b_2 + 2^1 b_1 + 2^0 b_0 \qquad (2)$$

where n_d is the decimal number, *n* is again the number of bits, and *b* is the condition of the bit in question starting with the most significant bit (MSB) and finishing with the least significant bit (LSB). A 4-bit binary code absolute encoder follows the binary code truth table in Table 1 below, which shows the decimal number and binary equivalent. The realization of this truth table

Table 1: Binary and gray code truth tables.

	Binary Code					**Gray Code**			
Sector	**MSB**			**LSB**		**MSB**			**LSB**
0	0	0	0	0		0	0	0	0
1	0	0	0	1		0	0	0	1
2	0	0	1	0		0	0	1	1
3	0	0	1	1		0	0	1	0
4	0	1	0	0		0	1	1	0
5	0	1	0	1		0	1	1	1
6	0	1	1	0		0	1	0	1
7	0	1	1	1		0	1	0	0
8	1	0	0	0		1	1	0	0
9	1	0	0	1		1	1	0	1
10	1	0	1	0		1	1	1	1
11	1	0	1	1		1	1	1	0
12	1	1	0	0		1	0	1	0
13	1	1	0	1		1	0	1	1
14	1	1	1	0		1	0	0	1
15	1	1	1	1		1	0	0	0

as an encoder is demonstrated in Figure 1 below. As can be seen in the figure, the inner track corresponds to the MSB and the outer track corresponds to the LSB. A white section is transparent, while a black section is opaque. If an infrared emitter/detector pair were situated astride each track, it would be possible to detect the absolute position of the disk within the limits of its resolution. The problem with this encoder is that it is possible for multiple bits to switch from one state to another at the same time. This is why a gray code absolute encoder was used for this exercise.



a. Binary code encoder b. Gray code encoder

Figure 1. Absolute encoder diagrams: (a) binary code and (b) gray code.

A gray code absolute encoder is similar to a binary code absolute encoder except that only one bit's condition can be changed at a time. The gray code truth table of Table 1 was provided to help in the design of the encoder. The encoder designed for the experiment is presented in Figure 1, which was then cut out of foam board in a disk of 7-in. diameter with 0.75-in. tracks. As can be seen, only one bit's condition could be changed as the disk rotated. Infrared detectors were fixed collinearly in foam board 0.75-in. apart. The IR sensors were then connected to a 5-volt power supply through 47 kΩ resistors to limit current, then tapped to the

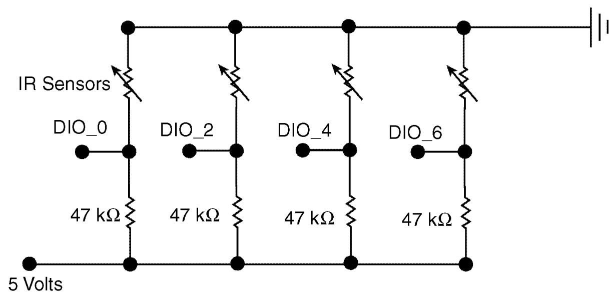


Figure 2. Circuit schematic.

digital inputs of the analog/digital board in the lab, as seen in Figure 2.

LSR was sent to DIO_0, bit 1 was sent to DIO_2, bit 2 was sent to DIO_4, and MSB was sent to DIO_6. The encoder was then fastened through its center to the foam board containing the IR sensors. An incandescent bulb was switched on to provide infrared light. The CVI project

ENCODER was run from the CVI/PROGRAMS/LABS folder. When the START button was pushed, the sector number was shown according to the gray code conversion, and it also showed the binary conversion of the binary number sensed.

In order to convert the gray code number to a binary number, the exclusive OR formula had to used as follows:

$$\begin{aligned} b_3 &= g_3 \\ b_2 &= b_3 \text{ XOR } g_2 \\ b_1 &= b_2 \text{ XOR } g_1 \\ b_0 &= b_1 \text{ XOR } g_0 \end{aligned}, \qquad (3)$$

where g_n is the n^{th} gray bit, and the exclusive OR follows the logic in Table 2 below:

Table 2: Exclusive OR logical truth table

b_n	g_{n-1}	b_n **XOR** g_{n-1}
0	0	0
0	1	1
1	0	1
1	1	0

Once a gray code number was converted to binary, equation (2) was used to convert it again to decimal. An example of this conversion is included in the Sample Calculations in the Appendix.

Results: The experiment was run as described in the methods. The resolution of the encoder was 22.5°. As each section was passed over the IR sensors, the results given by CVI were checked against the values given in Table 1 for both binary and gray codes, and found to be accurate. The experimental results were exactly as expected from the theoretical derivation set forth in the procedures.

Discussion: The encoder worked well, as expected. There is little that could have gone wrong in the experiment. If there were any error, it would have likely occurred because of trimming the edges of the encoder too close to where the IR sensor would have been located, resulting in light shining on the sensor instead of it being blocked. Another potential source of error would be the code written in CVI, particularly entering the XOR function properly.

This experiment clearly illustrated the potential of the absolute encoder as a rotary position sensor. The encoder used in the lab would be impractical since its resolution is so low. Its resolution could have been improved by adding extra tracks to the encoder, thus increasing the bit size. Each added bit of information would double the resolution. A drawback to this would be increasing the size and inertia of the disk. Indeed, for the absolute encoder to be useful, it would need to be very small, thus requiring smaller sensors.

The use of an incremental encoder would be a good option. The incremental encoder divides one track into a series of opaque and transparent sections like the LSB of the binary encoder. A computer or microprocessor keeps track of how many pulses go by, thus giving an accurate reading of how much rotation occurs. A second track 90° out of phase with the first provides an indication of the direction of rotation. Dividing the amount of rotation by the elapsed time provides the rotary speed. One problem with this setup is that the zero position floats between uses and must be reset. Sometimes a third band with only one notch cut out is added to the encoder to give an indication of the zero position. The resolution of the incremental encoder is limited only by the notch size, the size of the encoder, and the size of the light sensor.

Encoders could be useful to the robot project by providing information of the angular displacement and velocity of a wheel or mechanism. However, this information may be more practically obtained with a potentiometer.

APPENDIX

Sample Calculations:

Encoder resolution using equation (1):

$$
\begin{aligned}
res &= \frac{360°}{2^4} \\
&= \frac{360°}{16} \\
&= 22.5°
\end{aligned}
$$

Gray code to binary code to decimal number using equation (3) with Table 2, and then using equation (2):

Given gray code number

$$
\begin{aligned}
& g_3 g_2 g_1 g_0 = 1101 \\
& b_3 = g_3 = 1 \\
& b_2 = b_3 \text{ XOR } g_2 = 0 \\
& b_1 = b_2 \text{ XOR } g_1 = 0 \\
& b_0 = b_1 \text{ XOR } g_0 = 1 \\
& \therefore b_3 b_2 b_1 b_0 = 1001 \\
& \therefore n_d = 2^3 + 2^0 = 9
\end{aligned}
$$