**Reading the Encoder**

I. OBJECTIVES

The objective of this experiment is to demonstrate direction calculation of a servo motor by using its encoder and an Arduino.

II. EQUIPMENT

1. Elegoo MEGA2560
2. Aslong DC motor JGA25-371-12V-95RMP
3. Computer/laptop.

III. BACKGROUND

Servo motors, commonly referred as gearmotors, are electromechanical feedback loop devices used to command control of speed and position. They produce torque and velocity based on the supplied current and voltages. It is a geared motor that can be used to provide additional needed torque to perform certain tasks.

Servo motors are used extensively in industry, construction, and daily life. They are used in robotics, manufacturing, construction, and kitchen devices. Because of its extensive use, technologies that utilize these devices must be designed with safety and longevity in mind. With help from an encoder, servos can provide smooth control over the motor.

The gearmotor used in this experiment has six wires used to power the motor and obtain potentiometer signals using Hall sensor. The signal feedback inputs produce two square wave outputs that are typically 90 degrees out of phase, which is used to determine direction of the motor. The encoder resolution for this motor is 12 pulses per revolution (PPR), which is the number of high pulses for either square wave in one revolution. This information will be useful in the future when programming the device.

Diagram

Description automatically generated

Figure 1: technical drawing of the JGA25-371 motor.

1. *The encoder*

An encoder works by observing changes to a magnetic field created by a magnet attached to the motor shaft. As the motor rotates, the magnet induces a protentional difference caused by Hall effect sensors. The encoder has two outputs that trigger periodically, which switch between high and low logic values.

The switching order can be utilized in algorithms to determine and control speed and direction. Again, the gearmotor has two signal feedback, A and B, that outputs a potential difference when the motor is turned. If the motor shaft is turned counterclockwise, signal A is triggered high and B is an active low. If the shaft turns the other way, B is now high and A is low.

A picture containing shape

Description automatically generated

Figure 2: Trigger waveform when the motor shaft is rotated.

1. *Programming direction*

The trigger waveform allows the user to program direction. Negative and positive pulse counts can be determined the order in which A or B (the two square waves) rises. If A rises first, add +1, otherwise add -1 to the counter if B rises. Generally, direction is determined by the right-hand rule where the fingers curl in the direction of the rotation and the thumb extends outward. The direction is considered position when the thumb is in an upright position (“out of the page”) and negative when the thumb is pointed down (“into the page”).

Currently, the program counts pulse “directions.” Almost all applications measure position in terms of degrees, revolutions, or radians. The program must include conversions that is related to the model specifications, such as the encoder resolution and gear ratio of the gearmotor to obtain standard position units. This integrated encoder provides a resolution of 12 pulses per revolution which refers to the number of quadrature decoded states that exist between the two outputs A and B.

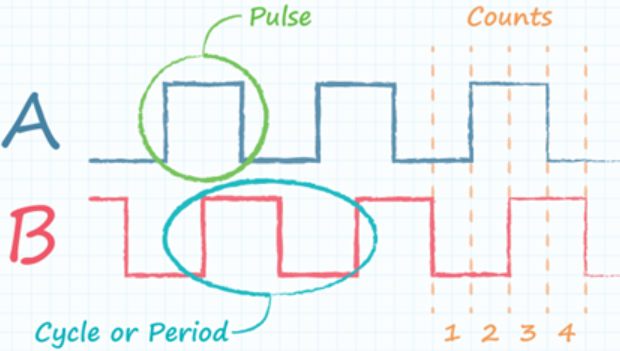


Figure 3: Pulses and counts of the two input signals.

Code can also utilize counts per revolution (CPR), which are two bits of information are represented as these four states. This may be a more useful method for PLC programming. It divides the PPR into four segments, outputting two bits of information for each state. The gear ratio is 1:45 which is always expressed as *driven: driver* or *output: input*. The gear ratio will be used similarly to an amplifier gain in a circuit to convert counts to revolutions with the expression

1. *Elegoo Mega 2560 R3*

The Elegoo Mega 2560 R3 is a microcontroller that is very close to the Arduino Mega 2560 Rev3. It is compatible with the Arduino IDE. Only power and digital (PWM output) pins and USB will be utilized. Digital pins will be used to read signals that will be used to calculate direction. The power pin port is the nexus between the Elegoo and the gearmotor (and later the motor driver needed to run the motor).

The USB connection provides communication channel between the MCU and personal computer. The speed at which communication is shared is controlled by the baud rate, or a measure of the number of changes to the signal (per second) that propagate through a transmission medium. Although baud rate is used by the Elegoo, bit rate is also commonly used in serial communication.

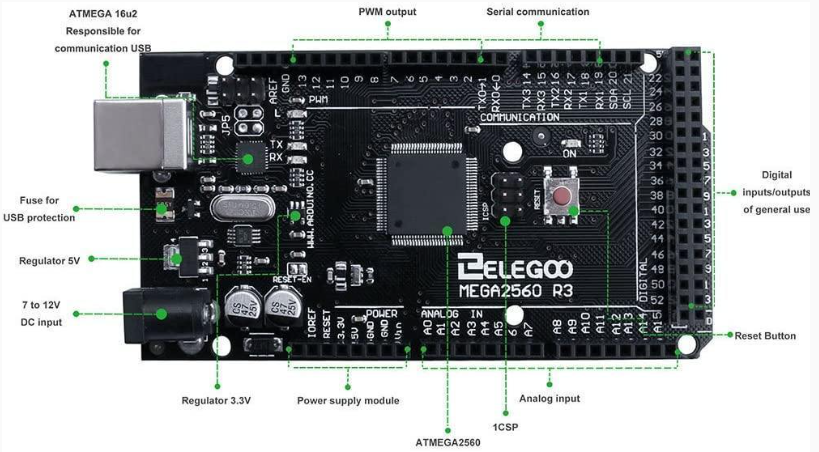


Figure 4 Pin and component layout for the Elegoo Mega 2560 R3.

1. *Programming the gearmotor*

The program can be separated in order by four parts: global variables, user created functions, definitions and initial states, and iterations. Global variables are utilized when the user wants the variable to be seen by the following three functions. User created functions aren’t always necessary but can be useful to keep the code organized and simple. The setup() function will only run the code once. It’s useful when defining pins and initial states. The loop() function reiterates over and over again which is necessary to update position and speed for control.

The attachInterrupt(digitalPinToInterrupt(pin), ISR, mode) function can be used to make encoder counts. The attachInterrupt has three functions: encoder output pin, the interrupt service routine function to call when the interrupt happens, and trigger type. The ISR is the most important feature as it will be code used to count our triggers. Instead of creating two interrupts to observe the two encoders.

The code can be simplified by observing just one encoder. If the encoder of interest is high, we can add up. If it is low, then it can be safely assumed the other encoder is high and we can subtract one from the total. The loop function was used to record and report the current position.

The program can be checked for bugs by clicking on *Verify* (checkmark) before uploading it with the *Upload* (right arrow) button.

IV. PROCEDURE

1. *Connecting the gearmotor and Arduino*
2. Connect the yellow wire to pin 2.
3. Connect the green wire to pin 3.
4. Connect blue wire to 5V.
5. Connect the black wire to GND.

A close-up of a circuit board

Description automatically generated with medium confidence

Figure 5: Pin connections of the JGA25-371 and Arduino.

1. *Connecting the Arduino to the interface*
2. Make the USB connection on the Arduino and open the Arduino IDE.
3. Save the sketch as *Lab1\_Measuring\_Direction*.
4. Click on the board drop box.
5. Under the Boards, select Arduino Mega or Mega 2560.
6. Under Ports, select COM3 Serial Port (USB).
7. *The program*

*Graphical user interface, text, application, email

Description automatically generated*

Figure 6: Program used to find position.

V. QUESTIONS