Lab 7: Final Project RPM Motor Control

Nate, Jack Tonkiga

1. Objective

The goal of this project was to acquaintance oneself more fully to some of our core concepts in embedded systems, otherwise known as a control system loop which as we learned, consists of sensors of components and actuators. We chose to do this with a motor control system because this will be a substantial and investigated in detail. A general condition to note for this series, while not a PID controller, is that the system must be stable.

Our goal was to create a DC motor control system that could operate at 1000 RPM with adequately good error or LED blinking light for ease of acknowledgement. This means the system would be able to generate a clock (50 pulses per second) and a Schmidt trigger circuit or the DC motor will be controlled to reach the number of 50 pulses generated by the above combination of phenomenons and treasure. Additionally, we wanted to be able to change the speed of the motor by a factor of 2 (500 RPM) and be able to change the direction of the motor (CW and CCW), and at the end of the course we were using a LATCH signal that turn off the LED and MOSFET generator and begin the process of the motor slowing down to 0 RPM. Finally, we wanted to be able to change the speed of the motor while it was in motion, but not to a halt. The processed analog signal is then passed through an error signal amplifier and a protecting buffer before being sent to the motor. Additionally, we wanted to be able to change the direction of the motor while it was in motion. The output of this navigation is then sent to the control loop through the motor via a power transistor.

The motor control circuit/processor that will be investigated in detail are as follows:

- LED and MOSFET generator
- Schmidt trigger circuit
- 8-bit DAC
- Error signal amplifier
- Error signal amplifier

The constraints for the overall error control circuit can be seen below in Fig 1.

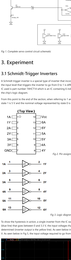
Motor
Error Signal Amplifier
Buffer
Power Transistor
Schmidt Trigger
Clock

Fig. 15 Analog counter operation

Fig. 16 Analog both digits on counter end

Fig. 17 Analog reset functionality with push button

What is the maximum that your counter can count to? Check what happens if you exceed this end device.

The counter can count to a maximum of 99 as based on the 7segment display, in case the counter exceeds the 99, it will reset back to 00. This exceeded the 7segment display relative to 99. The reference is similar to the 1 bit counter and so is the 2 digits counting from 00 to 99. The 7segment display is connected to the output of the counter and the reset button is connected to the reset input of the counter. The counter clock is also to each bit the capability to hold the end count as it is therefore by triggering a previous BCD 9 + 1 = 10 the counter will reset back to 00. The 7segment display is also connected to the output of the counter and the reset button is connected to the reset input of the counter. We have to a total of available bit and all available configurations of these bits connected you need to count from 00 to 99. The reason behind this the counter is to have 10 instead of decimal digits 9 and 6 instead from the same group of bits. 10 digits

Fig. 18 Traditional reset, "push" when LED#10 is triggered

The 74LS47 BCD decoder is used to drive the seven segment display, which is a common anode display. The 74LS47 has 16 inputs (labeled A to D, plus 14 outputs (labeled a to g). It allows one processor to interface with 16 BCD-to-seven. The ROM1 in the counter was grounded. Fig 19 below shows the 7segment display connected to the output of the counter and the reset button is connected to the reset input of the counter. Through there is some bug the image of the display or signals connected seem that the 7th bit is not working correctly.

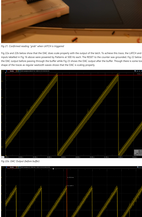

Fig. 19 Bit 7 Output Status LED#7

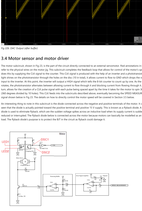

Fig. 20 Bit 7 Output Status LED#7

3.4 Motor sensor and motor driver

The motor sensor, shown in Fig. 21 below, is a type of sensor directly connected to an external environment. Then demonstrated in Fig.22 to refer to the physical state on the motor.pcb. The sensor.pcb completes the feedback loop that allows the control of the motor.l.capsule to adjust its operation based on the actual state of the motor. The motor.pcb is the physical component that converts electrical signals from the sensor.pcb into the mechanical force through the motor.l.m. Also the L-H-bridge, a device used to control the direction and speed of a motor by adjusting the voltage and current supplied to the motor, the motor.l.capsule is the physical enclosure that houses the motor.l.m and the L-H-bridge. Finally, the photosensor.pcb serves as the feedback mechanism allowing control to flow through it and blocking control from flowing through it. This is a crucial component in the system, as it provides the necessary feedback to the control system to ensure the motor operates as intended. Overall, these components work together to form a closed-loop control system, where the motor.pcb receives control signals from the sensor.pcb and adjusts its operation based on the feedback provided by the photosensor.pcb. The motor.l.capsule and L-H-bridge ensure that the motor operates within the desired parameters, while the sensor.pcb provides the initial signal that triggers the entire process.

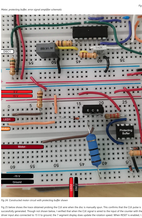

Fig. 21 Motor sensor and motor driver

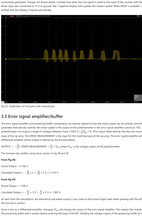

Fig. 22 Motor sensor and motor driver

3.5 Error signal amplifier/buffer

Fig. 23 below shows the error signal amplifier/buffer, which is a crucial component in the system, designed to ensure the motor operates as intended. The potentiometer that directly controls the motor speed is the output of the potentiometer is the error signal amplifier/buffer. The potentiometer is connected to the error signal amplifier/buffer, which amplifies the signal to ensure it is strong and stable before being sent to the motor.pcb. The ATMEL ATMEGA328PB-AU is the chip used for the reading input of the op-amp. The error signal amplifier acts as a buffer, isolating the potentiometer from the motor.pcb to prevent any interference or noise.

$$OUT = \left(1 + \frac{R2}{R1}\right) \times (IN2 - IN1) + IN1 \quad \left(1 + \frac{R_{feedback}}{R_{input}}\right) \times (V_{in2} - V_{in1}) + V_{in1}$$

The Excel sheet, available using more details in Fig.24 and 25.

Vmin Fig.24:

Output Equation = 1.0000

Calculated Output = (1 + (R2 / R1)) * (IN2 - IN1) + IN1

Vmax Fig.25:

Output Equation = 1.0000

Calculated Output = (1 + (R2 / R1)) * (IN2 - IN1) + IN1

As read from this accessories, the theoretical calculated output is very close to the actual output over when pushing with the ADC Nexus Pro 3 Oscilloscope.

Since it acts as a differential amplifier, changing R2 will not significantly change the output of the error signal amplifier. The output from Vmin Fig.24 is the potentiometer is the error signal amplifier/buffer. Binding the voltage output of the potentiometer to the Vin 1.25. Binding the voltage output of the potentiometer to the Vin 2.50. Adjusting the potentiometer will change the output of the op-amp. The ATMEL ATMEGA328PB-AU is the chip used for the reading input of the op-amp. The error signal amplifier acts as a buffer, isolating the potentiometer from the motor.pcb to prevent any interference or noise.

Connected to the op-amp is a voltage divider, which is used to scale the voltage output of the potentiometer to ensure it is within the op-amp's input range. The voltage divider is connected to the op-amp's input pins, ensuring that the voltage output of the potentiometer is scaled to a level that the op-amp is designed to handle. The op-amp's output is then connected to other circuit components, such as the motor.pcb, to control the motor's speed and direction. The voltage divider is connected to other circuit components, such as the motor.pcb, to provide the feedback signal to the op-amp. The op-amp's output is then used to adjust the motor's speed and direction based on the feedback signal from the voltage divider. The op-amp's output is also used to drive other circuit components, such as the motor.pcb, to control the motor's speed and direction.

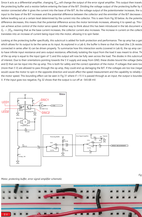

Fig. 23 Implementation of error signal amplifier/buffer

3.5 Error signal amplifier/buffer

Fig. 26 below shows the error signal amplifier/buffer, which is a crucial component in the system, designed to ensure the motor operates as intended. The potentiometer that directly controls the motor speed is the output of the potentiometer is the error signal amplifier/buffer. The potentiometer is connected to the error signal amplifier/buffer, which amplifies the signal to ensure it is strong and stable before being sent to the motor.pcb. The ATMEL ATMEGA328PB-AU is the chip used for the reading input of the op-amp. The error signal amplifier acts as a buffer, isolating the potentiometer from the motor.pcb to prevent any interference or noise.

Connected to the op-amp is a voltage divider, which is used to scale the voltage output of the potentiometer to ensure it is within the op-amp's input range. The voltage divider is connected to the op-amp's input pins, ensuring that the voltage output of the potentiometer is scaled to a level that the op-amp is designed to handle. The op-amp's output is then connected to other circuit components, such as the motor.pcb, to control the motor's speed and direction. The voltage divider is connected to other circuit components, such as the motor.pcb, to provide the feedback signal to the op-amp. The op-amp's output is then used to adjust the motor's speed and direction based on the feedback signal from the voltage divider. The op-amp's output is also used to drive other circuit components, such as the motor.pcb, to control the motor's speed and direction.

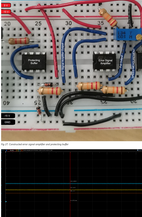

Fig. 24 Error signal amplifier/buffer

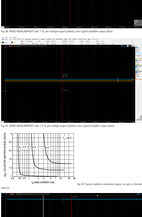

Fig. 25 Error signal amplifier/buffer

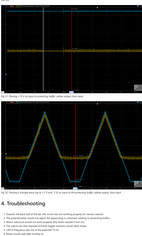

Fig. 26 Error signal amplifier/buffer

Fig. 27 Error signal amplifier/buffer

Fig. 28 Error signal amplifier/buffer

Fig. 29 Error signal amplifier/buffer

Fig. 30 Error signal amplifier/buffer

Fig. 31 Error signal amplifier/buffer

Fig. 32 Error signal amplifier/buffer

Fig. 33 Error signal amplifier/buffer

Fig. 34 Error signal amplifier/buffer

Fig. 35 Error signal amplifier/buffer

Fig. 36 Error signal amplifier/buffer

Fig. 37 Error signal amplifier/buffer

Fig. 38 Error signal amplifier/buffer

Fig. 39 Error signal amplifier/buffer

Fig. 40 Error signal amplifier/buffer

Fig. 41 Error signal amplifier/buffer

Fig. 42 Error signal amplifier/buffer

Fig. 43 Error signal amplifier/buffer

Fig. 44 Error signal amplifier/buffer

Fig. 45 Error signal amplifier/buffer

Fig. 46 Error signal amplifier/buffer

Fig. 47 Error signal amplifier/buffer

Fig. 48 Error signal amplifier/buffer

Fig. 49 Error signal amplifier/buffer

Fig. 50 Error signal amplifier/buffer

Fig. 51 Error signal amplifier/buffer

Fig. 52 Error signal amplifier/buffer

Fig. 53 Error signal amplifier/buffer

Fig. 54 Error signal amplifier/buffer

Fig. 55 Error signal amplifier/buffer

Fig. 56 Error signal amplifier/buffer

Fig. 57 Error signal amplifier/buffer

Fig. 58 Error signal amplifier/buffer

Fig. 59 Error signal amplifier/buffer

Fig. 60 Error signal amplifier/buffer

Fig. 61 Error signal amplifier/buffer

Fig. 62 Error signal amplifier/buffer

Fig. 63 Error signal amplifier/buffer

Fig. 64 Error signal amplifier/buffer

Fig. 65 Error signal amplifier/buffer

Fig. 66 Error signal amplifier/buffer

Fig. 67 Error signal amplifier/buffer

Fig. 68 Error signal amplifier/buffer

Fig. 69 Error signal amplifier/buffer

Fig. 70 Error signal amplifier/buffer

Fig. 71 Error signal amplifier/buffer

Fig. 72 Error signal amplifier/buffer

Fig. 73 Error signal amplifier/buffer

Fig. 74 Error signal amplifier/buffer

Fig. 75 Error signal amplifier/buffer

Fig. 76 Error signal amplifier/buffer

Fig. 77 Error signal amplifier/buffer

Fig. 78 Error signal amplifier/buffer

Fig. 79 Error signal amplifier/buffer

Fig. 80 Error signal amplifier/buffer

Fig. 81 Error signal amplifier/buffer

Fig. 82 Error signal amplifier/buffer

Fig. 83 Error signal amplifier/buffer

Fig. 84 Error signal amplifier/buffer

Fig. 85 Error signal amplifier/buffer

Fig. 86 Error signal amplifier/buffer

Fig. 87 Error signal amplifier/buffer

Fig. 88 Error signal amplifier/buffer

Fig. 89 Error signal amplifier/buffer

Fig. 90 Error signal amplifier/buffer

Fig. 91 Error signal amplifier/buffer

Fig. 92 Error signal amplifier/buffer

Fig. 93 Error signal amplifier/buffer

Fig. 94 Error signal amplifier/buffer

Fig. 95 Error signal amplifier/buffer

Fig. 96 Error signal amplifier/buffer

Fig. 97 Error signal amplifier/buffer

Fig. 98 Error signal amplifier/buffer

Fig. 99 Error signal amplifier/buffer

Fig. 100 Error signal amplifier/buffer

Fig. 101 Error signal amplifier/buffer

Fig. 102 Error signal amplifier/buffer

Fig. 103 Error signal amplifier/buffer

Fig. 104 Error signal amplifier/buffer

Fig. 105 Error signal amplifier/buffer

Fig. 106 Error signal amplifier/buffer

Fig. 107 Error signal amplifier/buffer

Fig. 108 Error signal amplifier/buffer

Fig. 109 Error signal amplifier/buffer

Fig. 110 Error signal amplifier/buffer

Fig. 111 Error signal amplifier/buffer

Fig. 112 Error signal amplifier/buffer

Fig. 113 Error signal amplifier/buffer

Fig. 114 Error signal amplifier/buffer

Fig. 115 Error signal amplifier/buffer

Fig. 116 Error signal amplifier/buffer

Fig. 117 Error signal amplifier/buffer

Fig. 118 Error signal amplifier/buffer

Fig. 119 Error signal amplifier/buffer

Fig. 120 Error signal amplifier/buffer

Fig. 121 Error signal amplifier/buffer

Fig. 122 Error signal amplifier/buffer

Fig. 123 Error signal amplifier/buffer

Fig. 124 Error signal amplifier/buffer

Fig. 125 Error signal amplifier/buffer

Fig. 126 Error signal amplifier/buffer

Fig. 127 Error signal amplifier/buffer

Fig. 128 Error signal amplifier/buffer

Fig. 129 Error signal amplifier/buffer

Fig. 130 Error signal amplifier/buffer

Fig. 131 Error signal amplifier/buffer

Fig. 132 Error signal amplifier/buffer

Fig. 133 Error signal amplifier/buffer

Fig. 134 Error signal amplifier/buffer

Fig. 135 Error signal amplifier/buffer

Fig. 136 Error signal amplifier/buffer

Fig. 137 Error signal amplifier/buffer

Fig. 138 Error signal amplifier/buffer

Fig. 139 Error signal amplifier/buffer

Fig. 140 Error signal amplifier/buffer

Fig. 141 Error signal amplifier/buffer

Fig. 142 Error signal amplifier/buffer

Fig. 143 Error signal amplifier/buffer

Fig. 144 Error signal amplifier/buffer

Fig. 145 Error signal amplifier/buffer

Fig. 146 Error signal amplifier/buffer

Fig. 147 Error signal amplifier/buffer

Fig. 148 Error signal amplifier/buffer

Fig. 149 Error signal amplifier/buffer

Fig. 150 Error signal amplifier/buffer

Fig. 151 Error signal amplifier/buffer

Fig. 152 Error signal amplifier/buffer

Fig. 153 Error signal amplifier/buffer

Fig. 154 Error signal amplifier/buffer

Fig. 155 Error signal amplifier/buffer

Fig. 156 Error signal amplifier/buffer

Fig. 157 Error signal amplifier/buffer

Fig. 158 Error signal amplifier/buffer

Fig. 159 Error signal amplifier/buffer

Fig. 160 Error signal amplifier/buffer

Fig. 161 Error signal amplifier/buffer

Fig. 162 Error signal amplifier/buffer

Fig. 163 Error signal amplifier/buffer

Fig. 164 Error signal amplifier/buffer

Fig. 165 Error signal amplifier/buffer

Fig. 166 Error signal amplifier/buffer

Fig. 167 Error signal amplifier/buffer

Fig. 168 Error signal amplifier/buffer

Fig. 169 Error signal amplifier/buffer

Fig. 170 Error signal amplifier/buffer

Fig. 171 Error signal amplifier/buffer

Fig. 172 Error signal amplifier/buffer

Fig. 173 Error signal amplifier/buffer

Fig. 174 Error signal amplifier/buffer

Fig. 175 Error signal amplifier/buffer

Fig. 176 Error signal amplifier/buffer

Fig. 177 Error signal amplifier/buffer

Fig. 178 Error signal amplifier/buffer

Fig. 179 Error signal amplifier/buffer

Fig. 180 Error signal amplifier/buffer

Fig. 181 Error signal amplifier/buffer

Fig. 182 Error signal amplifier/buffer

Fig. 183 Error signal amplifier/buffer

Fig. 184 Error signal amplifier/buffer

Fig. 185 Error signal amplifier/buffer

Fig. 186 Error signal amplifier/buffer

Fig. 187 Error signal amplifier/buffer

Fig. 188 Error signal amplifier/buffer

Fig. 189 Error signal amplifier/buffer

Fig. 190 Error signal amplifier/buffer

Fig. 191 Error signal amplifier/buffer

Fig. 192 Error signal amplifier/buffer

Fig. 193 Error signal amplifier/buffer

Fig. 194 Error signal amplifier/buffer

Fig. 195 Error signal amplifier/buffer

Fig. 196 Error signal amplifier/buffer

Fig. 197 Error signal amplifier/buffer

Fig. 198 Error signal amplifier/buffer

Fig. 199 Error signal amplifier/buffer

Fig. 200 Error signal amplifier/buffer

Fig. 201 Error signal amplifier/buffer

Fig. 202 Error signal amplifier/buffer

Fig. 203 Error signal amplifier/buffer

Fig. 204 Error signal amplifier/buffer

Fig. 205 Error signal amplifier/buffer

Fig. 206 Error signal amplifier/buffer

Fig. 207 Error signal amplifier/buffer

Fig. 208 Error signal amplifier/buffer

Fig. 209 Error signal amplifier/buffer

Fig. 210 Error signal amplifier/buffer

Fig. 211 Error signal amplifier/buffer

Fig. 212 Error signal amplifier/buffer

Fig. 213 Error signal amplifier/buffer

Fig

5. Conclusion

In 2021, the Pair did also highly complete and built on the knowledge we had obtained in previous labs to build and troubleshoot the circuitry, and we successfully completed the assignment and achieved the following results:

1. Part 1: Learning the Schmitt trigger inverter module to learn after Schmitt trigger the related circuits, although I had very nearly shorted once,it is better and confirmed that the 5 with the inverter was working before building the advance.
2. Part 2: Learning the 555 timer module, learning the 555 timer module, learning the 555 timer module, learning the 555 timer and some high-frequency circuit design, and successfully completed the ceramic to our circuit.
3. Part 3: Learning the 74HC193 counter module, learning the 74HC193 counter module, learning the 74HC193 counter module,