Lab 1 Questions

Q1.

How ADC works in Arduino.

The voltage is read on the pin, the digital value after conversion is from 0 to 1023 in 10-bit int format.

According to datasheet, in Mega2560, the 0 (0x000) to 1023(0x3FF) is equal to 0V to $\frac{1023}{1024}V_{ref}$ (which is $V_{ref} - 1LSB$). Therefore, the conversion is $ADC = \frac{V_{input} \times 1024}{V_{ref}}$.

Q2.

According to the datasheet, the V_{ref} of ADC can be selected as AVCC, internal 1.1V or 2.56V and external AREF pin. While not setting and not input connect to the AREF pin, the default reference voltage will be the AVCC, which is 5V on Arduino Mega 2560.

It is not the most optimum use of ADC because the resolution on the range (0 to V_{ref}) is certain (10-bit, 1024 resolution). As the input is between 0 – 3.3V, with a 5V reference voltage, the resolution is reduced from 0.00322V ($V_{ref} = 3.3V$) to 0.00488V ($V_{ref} = 5V$). To increase the resolution, we can connect the AREF to a 3.3V stable external input as reference voltage.

Q3.

10% duty cycle:

Figure 1. PWM waveform of 10% duty cycle

25% duty cycle:

Figure 2. PWM waveform of 25% duty cycle

50% duty cycle:

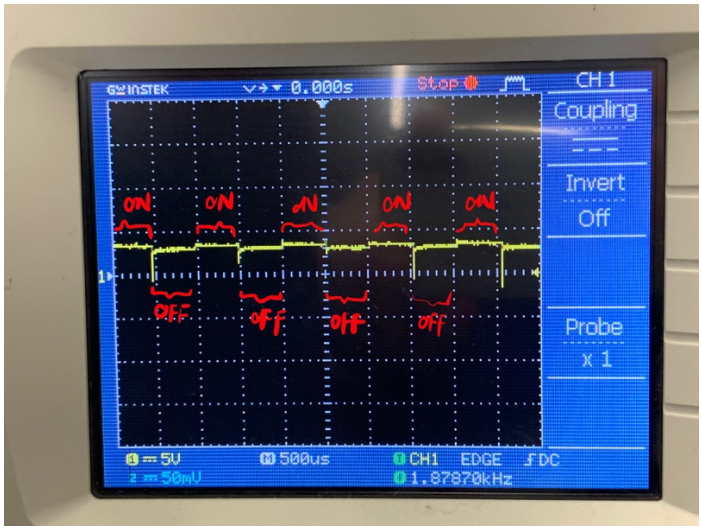

Figure 3. PWM waveform of 50% duty cycle

75% duty cycle:

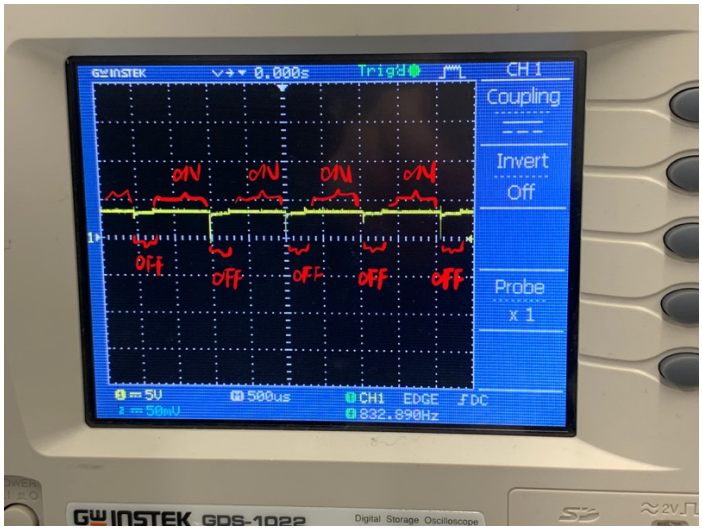

Figure 4. PWM waveform of 75% duty cycle

100% duty cycle:

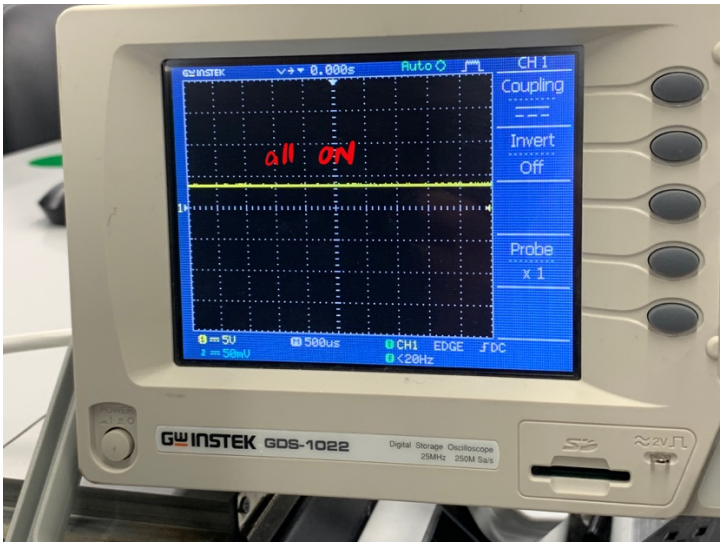

Figure 5. PWM waveform of 100% duty cycle

When it is setting to 100%, the display on oscilloscope is a flat line with all the time at high voltage.
When output at high frequency, the impedance might affect the final output and leads to non-zero output at low PWM output.

Q4.

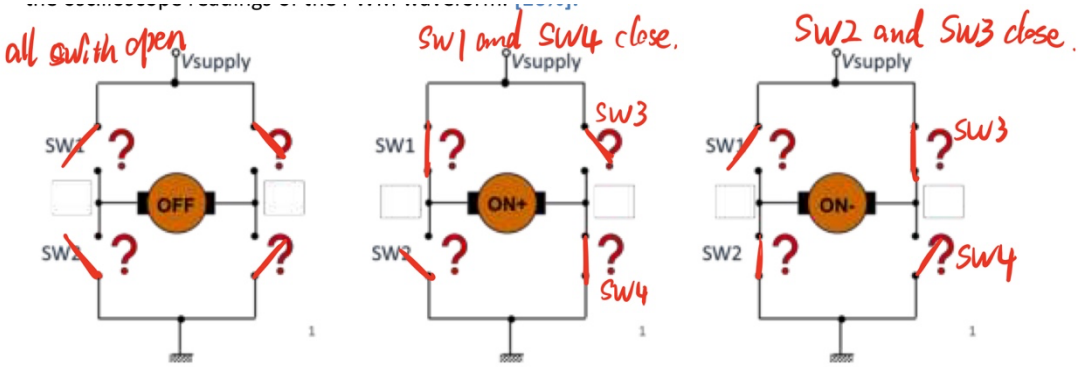

Figure 6. Switch configuration of H-bridge circuit diagram

According to datasheet, default PWM frequency of Arduino Mega is 490Hz for all pins, except pins 4 and 13 are 980Hz. The pin 4 and pin 13 are using timer 0, which use a different pre-scaler setting.
With 75% duty cycle, it already provides an audible note, the frequency is around 800Hz.
Evidence:

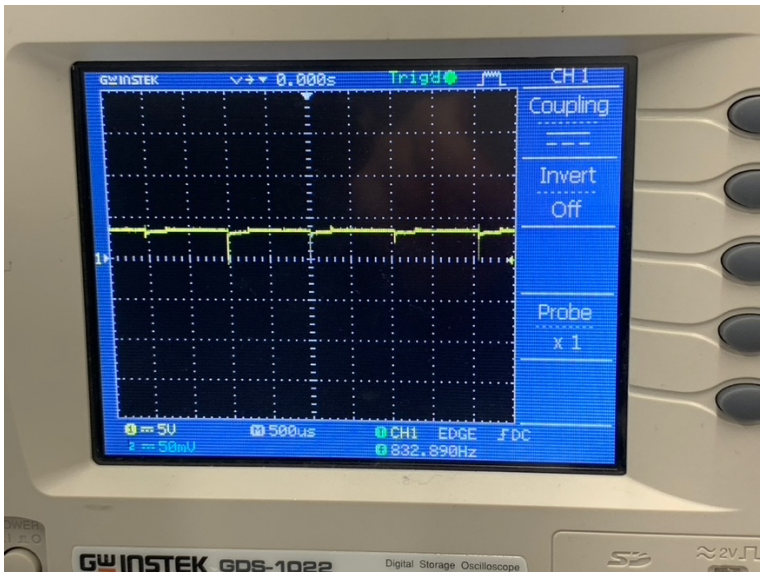

Figure 7. the oscilloscope readings of the PWM waveform with audible note

Q5.

Positive:

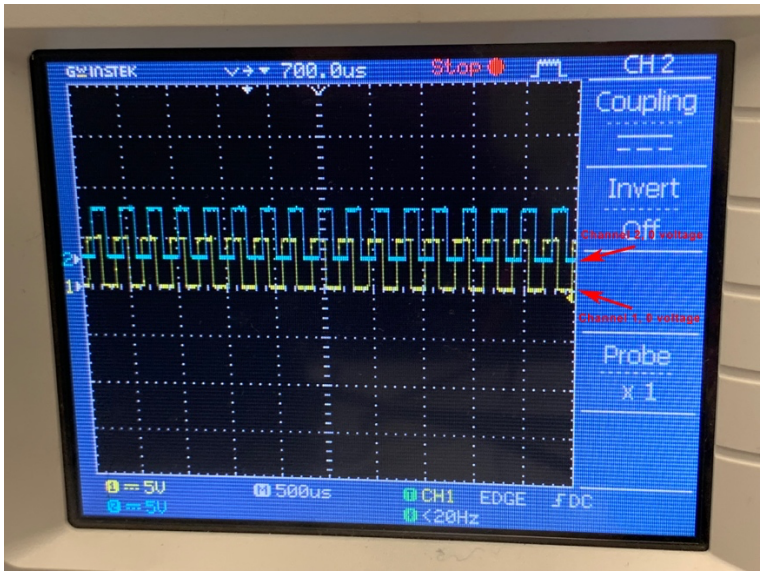

Figure 8. Quadrature Signals (Oscilloscope) when running in positive direction

Negative:

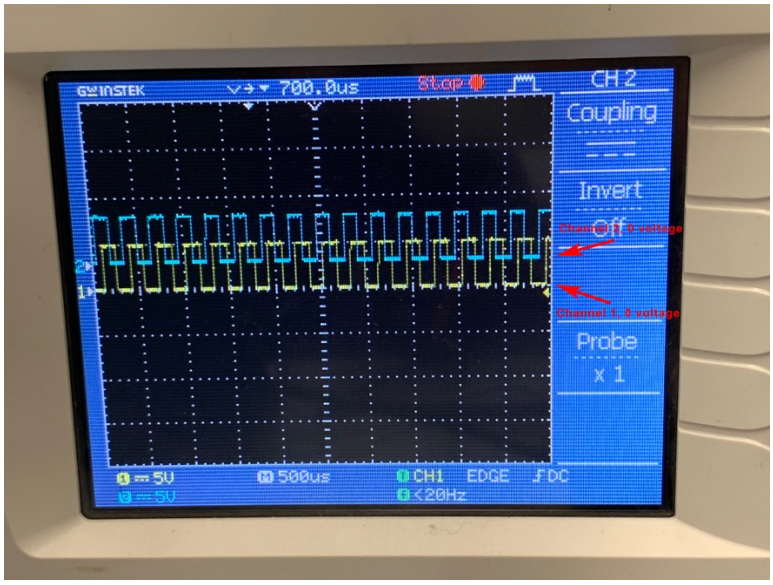

Figure 9. Quadrature Signals (Oscilloscope) when running in negative direction

Q6.

No, it can't be used as an alternative to the LS7366R. Although the counting is quite accurate, but only a quarter of pluses is countered. This already leads to the error between the LS7366 and internal while changing the direction of the motor.

Q7.

Some errors are counted during the high-speed rotation of the motor. This might be due to the unfinish of the pervious interrupt, but the external source changed and toggle the next interrupt and lead to the error.

Q8.

A control loop can be used to make the motor achieve accurate positioning. The current position can be either detected by a sensor or using integration and start point to maintain a rough position.

Listing of the Programs

TwoSensorsCZ.c
: C program with console interface for unit test on program logic.

TestEncoderCZ.c
: C program with console interface for unit test on program logic.

TwoSensorsCZ.ino
: The Arduino code for the exercise 1, interfacing with thermocouple and thermistor.

TestEncoderCZ.ino
: The Arduino code for the unit test of sample state machine functions.

MotorEncoderCZ.ino
: The Arduino code for the exercise 2 and 3.