**Lab 2 Report**

Jonathan Monreal (jem177) and Jiawei Wu (jxw585)

Group 8

**Design Questions**

1. ***What is the control signal frequency specification of the motor driver used on the daughter board?***

Typ. 30KHz, Max. 100KHz

1. ***What is the chosen output frequency of your PWM generator?***

It is ~25kHz

1. ***Assuming a maximum speed of 6000 rpm, what is the maximum motor speed your design will calculate with the provided 192 count/turn encoder attached to the motor?***

The maximum speed our design will calculate is ~3200 rpm

1. ***How frequently does your design determine the motor's rate of rotation?***

The reset frequency is ~12KHz.

1. ***Calculate approximately how long it takes to change the speed goal from 0 to full speed in one direction if the button is held constantly.***

It takes about 40 seconds for it to saturate at a maximum speed from zero.

1. ***At maximum gain, what is the pulse width when the goal is minimum and maximum for the free-running, enabled motor?***

The maximum pulse width is the input clock period of the pwm, which is ~0.04ms

The minimum pulse width is 0, since minimum goal is 0.

**Counters**

***goal-counter***

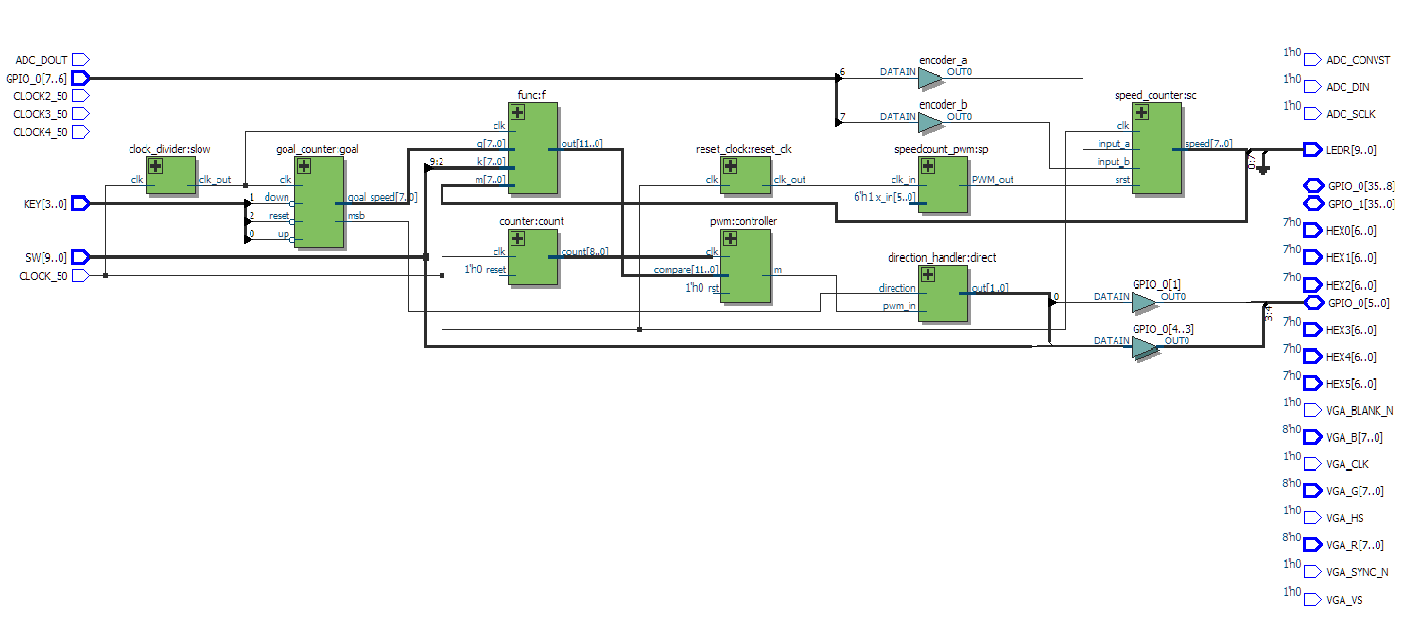
The goal counter is a counter that is associated with the speed up buttons of two directions. It counts an absolute value of 8 bit binary continuously from pressing down the buttons. The direction is handled using a dedicated module.

***speed\_counter***

The speed counter is the counter of measuring the feedback speed read from the encoder, which counts up every time the motor passes a groove, and it gets reset at a clock frequency, ensuring a measure of speed. The measurement is used in the equation of r = k(g-m); feedback speed is also mapped to the LEDs on board.

**Code Organization**

We followed the code organization so that the codes satisfy the requirements.

**Design Diagram and Module Functionality**

**Module clock\_divider** is a clock of low frequency that is associated with the buttons press trigger rate.

**Module goal\_counter** is the counter for goal speed from the button press, it generates a direction that is handled separately.

**Module func** is the implementation of the equation r=k(g-m) that we would use.

**Module pwm** determines how much duty cycle the motor would receive.

**Module direction\_handler** takes the direction bit from goal and determines which way the motor turns.

**Module speedcount\_pwm** is a pwm adjusting the reset/count percentage of the clock used to measure feedback speed\

**Module speed\_counter** is the motor speed feedback measurement

**Oscilloscope Captures**

**Collaboration**

We write draft codes and using github to exchange, working in lab together to debug.