Project Title: *Digital Angle Gauge*

Digital Angle Gauge (DAG) is a device used to set the saw blade angle in a wood shop. It is stuck to the saw blade and the angle of the saw blade is adjusted until the value shown by the DAG matches with the required value.

Objective:

I will be implementing a Digital Angle Gauge device using the KL25Z. At the core of our DAG, we shall be using the built-in MMA8451Q Accelerometer sensor and interface it with the KL25Z over I2C.

Functionality:

Before Command is entered.

Before the user enters a command (before entering process_command()), the command_line_interface() will be non-blocking such that it indicates if the accelerometer is perfectly flat on the table using red-color LED. The color will be varied using PWM to indicate to the user that orientation of the DAG has changed and is not flat.

After Command is entered: (The following commands are blocking)

1) **Calibrate Command:** Before using the DAG, the accelerometer must be calibrated in such a way that the current angle is considered as 0°. This will be done as shown below:
   - The user will send a command over the command processor to enter the calibrate mode.
   - The user will press the switch to finalize the relative zero-degree angle.

   Before the DAG is calibrated the white LED is Lit. Once the user has configured the DAG to relative 0° angle, Green LED will be lit(no PWM) along with a comment on the terminal.

   If the DAG is not calibrated and the user presses the SET command, the relative zero-degree angle is considered as 0 and the green LED is lit.

2) **Set Command:** The desired angle value will be entered by the user through the command processor. (Set <angle> for example, set 90)
3) LEDs will be used to indicate the current angle of the DAG. The color of the LED will be modified using PWM. The color will vary from **green** to **blue** using PWM as the desired angle is reached.
4) Once the user has tilted the DAG to the desired value, Blue LED will be lit along with a comment on the terminal. The user **MUST** press the touch sensor to exit this command handler.
   Calibration must be done again to set the desired angle. If no calibration is done, the relative zero-degree angle is set to actual 0°.
5) Additional Commands (**help command**) on the Command Handler will give information on the current angle.

Command	Usage
Help	Prints out info about all commands
Info	Prints out relative zero and angle w.r.t. to Relative zero
Calibrate	Used to set the relative zero
Set	Used to set the desired angle
Author	Prints out the name of the author

Technologies:

1) Command Processor
    - Interrupt based UART
    - Circular Buffers
2) MMA8451Q Interface
    - I2C
3) Interrupts
    - GPIO interrupt – Switch
    - Touch Sensor Interrupt
4) PWM

Anticipated Learnings and Demonstration of deeper Knowledge:

1) The command_line_interface() needs to be non-blocking so that it lets the user know if the device is flat on the table.
2) The MMA8451Q must be interfaced with the KL25Z board using I2C protocol. Hence, I will learn more about the I2C protocol and interfacing using I2C.
3) Conversion of MMA8451Q values read by the sensor into angles.
4) There are multiple interrupts present and all these interrupts must be handled correctly.
5) Testing of various aspects of this project like:
    - I2C
    - PWM
    - MMA8451Q
    - Command Processor
    - Interrupts
6) Integrating various technologies into a single project. This will help in writing clean code.

Sources for References:

1. For the MMA8451Q interface, I will be referencing the data sheet in the link shown below: (I2C characteristic, System modes, Serial I2C Interface, etc). https://www.nxp.com/docs/en/data-sheet/MMA8451Q.pdf
2. KL25Z reference manual for I2C interface.

Hardware Requirements:

- Switch – GPIO interrupt

Testing Strategy:

Automated and Semi- Automatic Testing:

1. CBFIFO testing - Fully Automatic
2. MMA8451Q testing – Semi- Automatic. (Will require the user to follow instructions shown on the terminal)

Manual Testing:

1. I2C characteristics testing
2. PWM working testing (LEDs)
3. Interrupts testing
4. Command Processor testing – Different commands

Hardware Block Diagram: