Digital Angle Gauge

Introduction

Digital angle Gauge is a device used to measure angle of something with respect to calibrated 0^o angle. The device is mainly used for tuning saws (although it can be used for any other purposes of angle measurement). The digital angle gauge will be developed using the MMA8451Q inbuilt accelerometer in FRDMKL25Z board. The sensor is low power, three axis and capacitive with 14 bit resolution. The 14 bit resolution of the sensor will be enough for my project and will give good accuracy for the application.

Functionality

The Digital Angle Gauge will show in real-time what the angle is with respect to the calibrated angle. The data will be shown on the serial terminal. The user can calibrate the sensor by pressing button attached to the microcontroller or my typing a command in the terminal, this will generate an interrupt and will calibrate the sensor to current value as 0^o. The user ca also use the touch sensor to calibrate the gauge.

In addition to this the user will get visual feedback about the device orientation using the onboard LEDs. There will be other commands in the command processor to help the user with using the device (like what does each color of light indicate, how to calibrate the sensor, etc).

The LED color changing is gradual (like we did in Buffahiti), PWM is used to make this happen.

Technologies Used

Some of the major technologies used are,

- Circular Buffers (for command processing)
- State Machine
- I2C (for sensor interfacing)
- Command Processing
- GPIO lines (for GPIO interrupts)
- Interrupts (for calibration)
- Pulse Width Modulation (for LED color changing)
- UART (for command processing)

Demonstration of Deeper Knowledge:

Command processing will have challenges as the functionality is different from what we have done for assignment, when user types “display” command the raw value from the sensor is shown on the terminal. When the user types “set0” command, the calibration will be done for the sensor.

Command processor is not the only one printing to terminal, there are other functions also printing on the terminal.

There will be multiple interrupts triggering the calibration, the user can trigger the sensor in two ways,

1. Using the switch attached to GPIO pins
2. Using “set0” command on the terminal screen

Since all of these are interrupts, they need to be given priority accordingly.

The code must be designed such that, all modules are independent as the user can calibrate any angle as 0. This will be a huge challenge as everything needs to be designed on the calibration point.

Circular buffer is tested for working in Breakfast serial assignment but will be adding more test cases as I have some more edge cases in mind.

I2C is a new area I will be working on in this project, I will understand the protocol and data communication by reading about it and use the learning to interface the sensor.

I will enjoy doing this project as it is combination of many things we learned in class.

Anticipated Learnings:

The following are the anticipated learning for the successful completion of the project,

1. I2C protocol in detail
2. MMA8451Q sensor and working.
3. Interrupt priority and handling
4. Multiple command processing and control of data flow using commands.
5. New state machine design as the user can change calibration values.
6. Modular coding style – to make the code not dependent on any of the sensor values, but only on the users calibration point.

I would be referring to the following documents for learning,

1. ARMv6-M Architecture Reference Manual
2. FRDM-KL25Z Schematics
3. KL25-Z Sub family reference manual.
4. MMA8451Q Datasheet or understanding

Additional Hardware:

No additional hardware is required except external switch which is very easy to procure. The code and working can be tested on any FRDMKL25Z board as I will be using the inbuilt Accelerometer sensor.

Testing strategies:

Since there are lot of things to be tested on this project, both manual and automated tests will be done to ensure correct working. The following are some of the test plans,

Automated tests :

1. The circular buffer will be tested for many cases.
2. Communication and initialization of MMA8451Q sensor tests.
3. LEDs will be tested for working by blinking all the colors and PWM mode.
4. Calibrated value test(Without movement).

Manual tests:

1. I2C communication tests for verifying functionality.
2. Testing GPIO interrupt using the switch.
3. Testing command processor for working valid and invalid commands.
4. State machine working will be tested by running different scenarios.
5. All the interrupt working verification by giving two or more interrupts together.
6. Testing the angle measured using semi-automated test (by tilting the sensor)

Some Commands in the command processor

Command	Function
Author	Will print the author of the project
Set0	Sets the current value measured as 0^{o}
Help	Details about how to calibrate and use the device

Sample State Diagram that I am planning to implement,