ECEN 5813 – Principles of Embedded Software
Course Project Proposal

Digital Angle Gauge using a 3-Axis accelerometer

TABLE OF CONTENTS

Section 1 - Project Proposal

Section 1: Objective and Project functionality

1.1 Technology Overview

Since time immemorial, man has been cutting wood, and he has also struggled to find a reliable way to cut precise angles. Power tools have definitely made the cutting easier. However, they have not improved the accuracy in any way. The blade tilt scales on most table saws are notoriously inaccurate and hard to read. A digital angle gauge provides an inexpensive and efficient way to monitor complex angles to make precise incisions.

This solution employs the MMA8451Q accelerometer on the FRDM KL25Z board to measure accurate angles. A 'zero button' is provided to calibrate the angular output to 0 degrees to a reference surface to provide the desired accuracy. The system also employs the LED subsystem controlled by PWM outputs that will be used to indicate the tilt angle. The system will output calibrated accelerometer readings over a serial UART port for accurate debugging.

1.2 Subsystem Analysis

The system is primarily a Digital angle gauge, used in order to measure complex angles for the user to set a saw blade on a table to saw exactly 45°. The LED subsystem will be used to indicate changes in angles.

1. The project will employ the on-board 3-axis accelerometer MMA8451Q interfaced over an I2C interface. The MMA8451Q will also employ Wake/Sleep mode in order to reduce power consumption by sleeping when the sensor is not in use. The accelerometer subsystem will be calibrated to 0° by pressing a button when it is placed on the reference surface.
2. The LED subsystem will be driven by the PWM module and will change colors according to the angle output. At 45°, the color on the LED will resolve to **Green** and blink twice to indicate that the desired angle has been achieved. This will be the subsystem used for indications. If there is no motion detected, the LEDs will be disabled to conserve power.
3. A GPIO interrupt will be used as an input to the system from a button that will calibrate the accelerometer in reference to the surface it is placed on.
4. A serial UART port will be used to display the read values in order to debug the system.

1.3 High Level Design

The high-level design section will outline the modes that the device will operate in, with a flowchart describing the flow of the operation.

- Power-on: The device will first be powered-on and the accelerometer will be woken up from sleep mode, if not already.
- Calibration: The device operation will rely on calibrating the accelerometer to reflect accurate angle values. After power-on, pressing a button will calibrate the device in reference to the surface it is placed on to reflect 0 values.

- After calibration, tilting the accelerometer will cause a color change in the RGB LED. When woken up from sleep mode, the accelerometer will update values every 100msec.
- Device Operation: Upon reaching the desired angle, the LED will resolve to a green color and blink twice to indicate that the angle has been reached.
- Sleep mode: When the device is not in use, the I2C slave – MMA8451Q accelerometer will be in sleep mode to conserve power. When the accelerometer is not in use, the LEDs will also be disabled.

This process is repeated, but the device does not need to be powered on each time.

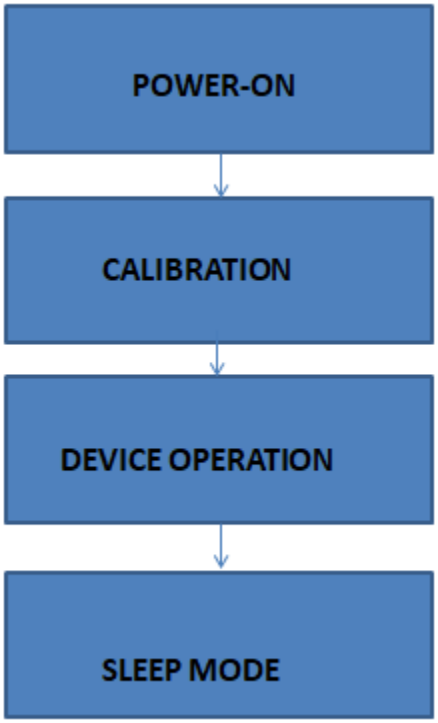


Figure 1 – Process Flow Diagram

Section 2: Hardware and Software Block diagrams

2.0 Functional Hardware Block Diagram

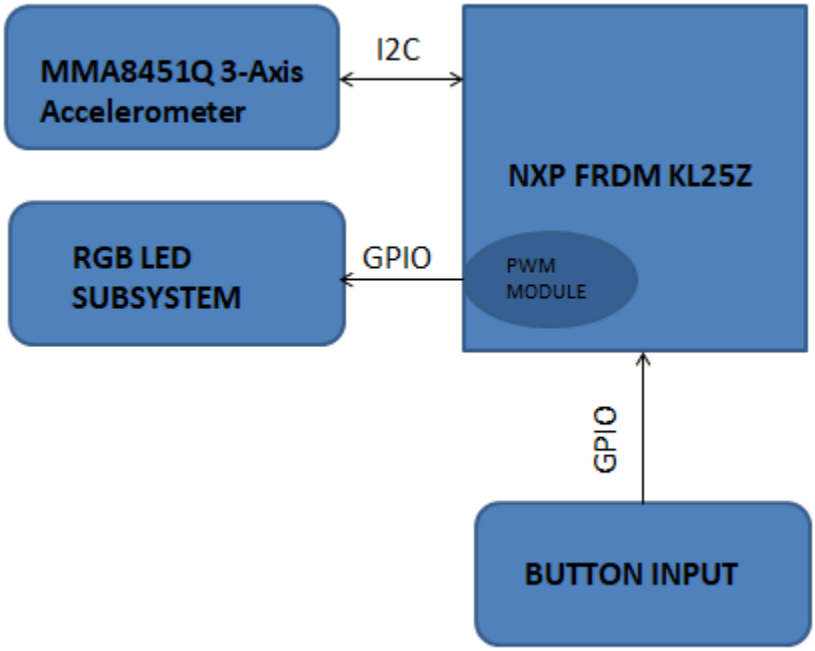
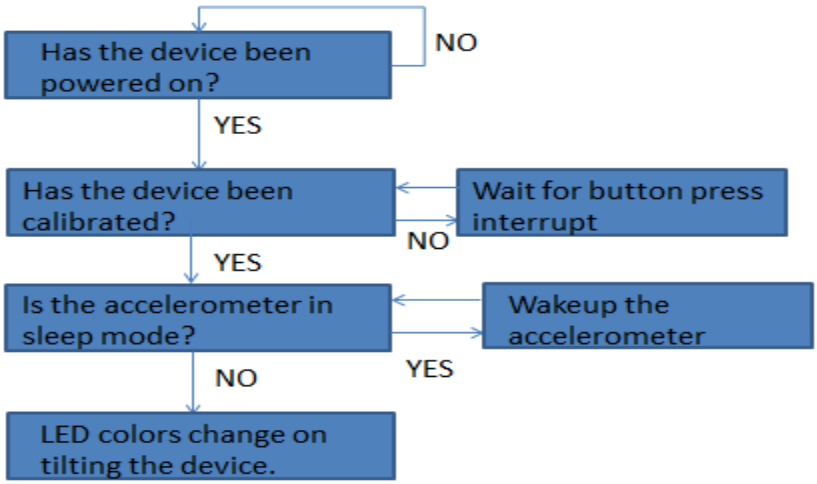


Figure 2 – Hardware Block diagram

2.1 Functional Software Flow Diagram

The software block diagram below will describe the software flow for the device.

Figure 3 – Software Flow diagram



Section 3: End-user functionality and Testing

3.1 End-user functionality

The end-user of the device should expect the following behavior:

- The RGB LEDs will be turned off when the accelerometer is in sleep mode.
- When the accelerometer is woken up from sleep, the RGB LEDs will glow based on the tilt angle.
- The device will then need to be placed on the reference plane and the user will press a button in order to calibrate the device.
- Once the device is calibrated, tilting the device will cause a change in the LED colors.
- When the desired angle (45°) is reached, the LED resolves to green color and blinks twice.
- If no motion is detected, the accelerometer goes to sleep mode. The LED subsystem is also disabled when there is no motion detected.

3.2 Testing

The testing protocol for the device involves a series of manual tests and automated tests. The test suite will cover a wide range of testing conditions. The final project submission will contain details about the test suite and the test results.
Some of the tests that will be included are:

- I2C scanner test: This test will check for the I2C slave on the device. (Automated)
- UART test: This test will contain tests for the circular queue and the UART module. (Automated)
- RGB LED color tests (Manual)
- Accelerometer testing (Manual).

Section 4: Learning outcome

This project will result in using the following software techniques:

- State machines
- I2C
- UART
- PWM
- Interrupts
- Clock configuration
- Register configuration
- Memory considerations
- Power considerations
- Reference Manual look-up

The I2C module is a new concept from an educational standpoint and will warrant having to spend some time learning it. All technologies used will use the best software practices and be cognizant of memory use as well.

Section 5: Hardware requirement

The system requires a button to be interfaced to the GPIO module in order to calibrate the accelerometer. No other external hardware is required.

Section 6: Github link

GITHUB REPOSITORY: Click here for Github Repo