Fall 2021 ECEN 5823

Course Project Proposal

Topic: Pedometer using 3-Axis Digital Accelerometer

By: Ruchit Naik

Contents

Figures

Project Definition

Developing a pedometer to detect the number of steps accurately and display the count on the 16x2 display. Also, configure a GPIO interrupt so that MMA8451Q (Accelerometer) can notify the KL25Z to start counting number of steps.

Objective

- Interfacing 3-axis accelerometer (MMA8451Q) with KL25Z over I2C to measure the movement across all the 3-axis.
- Develop an algorithm to detect the number of steps accurately based on the reading from the accelerometer.
- Interface 16x2 LCD to display the out of number of steps for the user.

Features

A pedometer calculates the total number of steps taken by the person. While taking the steps, there may be variation in motion in all 3-axis. The pedometer, by its core, uses measurements of acceleration in the direction of all 3 axis.

While running the pedometer can be in any orientation, so it is necessary to incorporate all acceleration in all the 3-axis. To do so, we would calculate the mean of square of acceleration in 3 axis.

In order to get the complete functionality of the device we would implement all the following features:

- It counts the steps i.e., number of changes in acceleration in any orientation.
- Displays the number of steps registered on the LCD
- The touch panel or the external button is used to reset the pedometer reading to 0.
- The button is interfaced as an external GPIO interrupt.
- The device should take the initial reading only if accelerometer based external interrupt is generated.

System Block Diagram

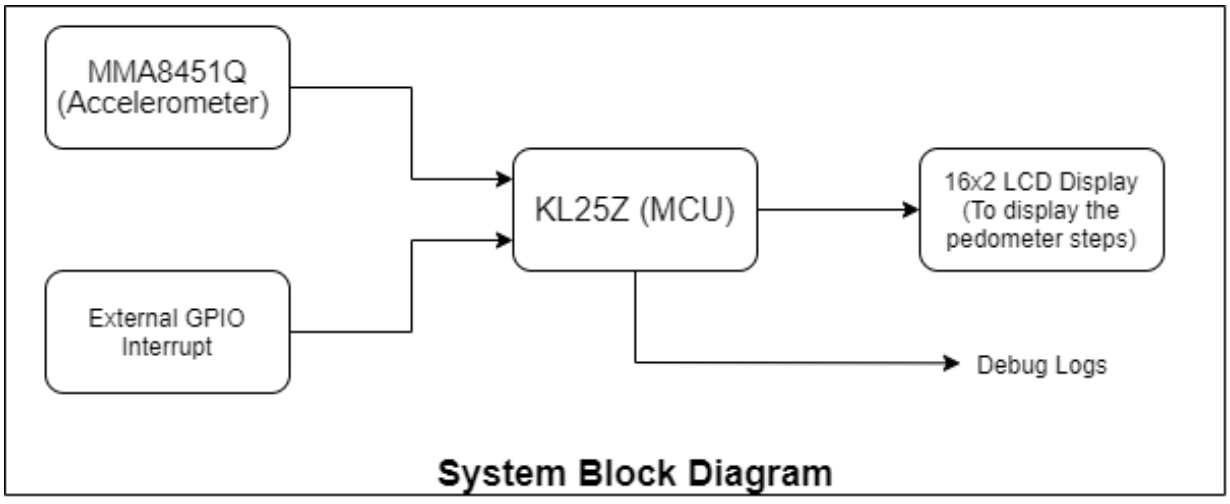

Figure 1: System Block Diagram

Hardware Requirements

- FRDM-KL25Z Evaluation Board
- 5V 16x2 LCD
- External Push-Button/TSI Touch Sensor
- MMA8451Q - Accelerometer (On-board accelerometer)
- Header connectors for evaluation board

Technologies Used

The project used I2C to interface the sensor with the MCU. I propose to use I2C0 port over the GPIO port E pins to interface the sensor with KL25Z.

To interface the LCD with MCU, I propose to use the GPIO to write the memory registers for of the LCD. We would use DDRAM (Data Display Random Access Memory) to write the commands and characters on the display.

I also propose to use external GPIO interrupt to interface push-button in interrupt mode to reset the pedometer count. Also, MMA8451Q could be configured in interrupt mode to start reading only when interrupt is received.

Anticipated learning

The project would require me to learn the I2C architecture on KL25Z and its corresponding registers. It would also need me to be acquainted with the MMA8451Q data sheet to analyze the required registers to configure and initialize the sensor and read the measurements from the sensor.

To interface the 16x2 LCD with the MCU, it is necessary to study the LCD data sheet and DDRAM architecture. It would be necessary to initialize the LCD and display the required ASCII characters on the hardware.

To implement the interrupt, I also need to go through the documentation to seek guidance on how to configure NVIC for external interrupts and how to handle the interrupts using the specific interrupt handlers.

Required Reference Documents

- MMA8451Q 3-Axis, 14-bit/8-bit Digital Accelerometer, Data sheet (nxp.com)
- FRDM-KL25Z Schematics (nxp.com)
- Kinetis KL25: 48MHz Cortex-M0+ 32-128KB Flash 32-80 pin (nxp.com)
- ARMv6-M Architecture Reference Manual
- KL25 Sub-Family Reference Manual - NXP Semiconductors 2018-12-20¢ KL25 Sub-Family Reference Manual, - [PDF Document] (documents.pub)
- Interfacing 16×2 LCD with KL25Z Series MCU – Learning Embedded System and Microcontrollers (wordpress.com)

The above links provide reference to each of the anticipated learning sections that are necessary to implement the above listed technologies.

Software Design Flow Diagram

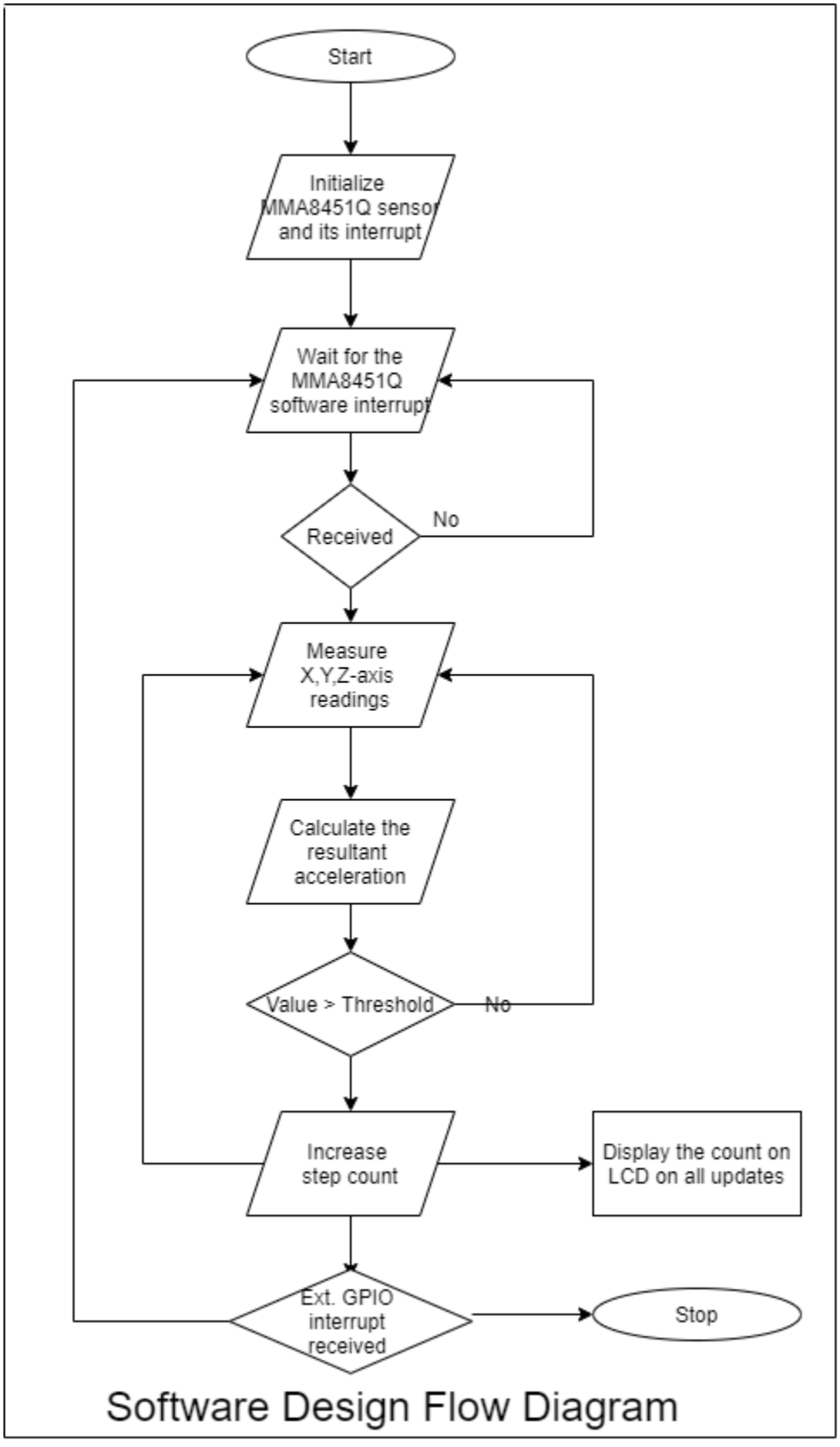

Figure 2: Software Design Flow Diagram

List of Software Modules

The list below mentions all software modules required to implement the objectives of the project. Each module would have .h and .c files

- **16x2 LCD library**: It would implement the library for the 16x2 LCD interfacing. It would handle LCD initialization and sending data to the DDRAM of the LCD hardware. It would also contain API to display the characters on the LCD.
- **MMA8451Q library**: It would handle all interfacing routine of the accelerometer with the MCU. It would handle to initialization of the sensor; write commands and read data to and from the sensor over I2C.
- **I2C library**: It would contain the bare-metal implementation to handle I2C mainly over I2C0 on the MCU. It would handle all routines to initialize, read and write over I2C.
- **GPIO library**: Bare-metal implementation to communicate over GPIO over MCU.
- **Interrupt Request Handling:** It would contain IRQ handlers for the necessary interrupts.

Goals

- To interface the accelerometer with KL25Z over I2C. Implement an algorithm to count the number of steps taken based on accelerometer readings. The goal is to achieve high precision to detect the steps.
- Interface 16x2 LCD with KL25Z to display the pedometer counts.
- Configure MMA8451Q to trigger interrupt on which the MCU should read the accelerometer.
- Configure a GPIO to external NVIC GPIO interrupt to interface a button to clear the pedometer count.

Stretched Goals

- Interface GPS sensor which would work in coherence with the accelerometer. It would help increase the accuracy to detect steps and avoid other unnecessary vibration and fake shaking of the device.
- Interface Wi-Fi module to send the step counts on to the server to display it on a web-based UI or a phone app.

Testing Strategy

The project would require manual testing to judge accurate function of the pedometer.

- Verify using logs if MMA8451Q is accurately interfaced and MCU gets the value when the eval board is moved.
- Verify visually if the LCD is interfaced properly and no garbage data is displayed.
- Test if change in MMA8451Q reading cause interrupts.
- Shake along X-axis of the sensor to check if a step is registered.
- Shake along Y-axis of the sensor to check if a step is registered.
- Shake along Z-axis of the sensor to check if a step is registered.
- Place the device on a flat ground to test if there are no steps registered on the pedometer display. – Check if error is handles correctly
- Test if push-button press clears the pedometer reading and restarts the step counting sequence.