NTNU – Trondheim
Norwegian University of
Science and Technology

TDT4258 Energy Efficient Computer Design
Laboratory Report

Exercise 1

Group 26:

Anders Åsheim
Fredrik Skretteberg
Håvard Dahle Løvaas

February 5, 2015

Abstract

An abstract is a short (100 to 500 words), high-level summary of the entire document. For this kind of report, you would start by introducing the concept that the report talks about and the goals of the work, followed by information about how the work was done and some summary of results.

1 Introduction

The goal of this exercise is to make a program that allows a user to control 8 LEDs with a gamepad with 8 buttons. And the microcontroller should have as little power consumption as possible. The program language that will be used for this exercise is assembly, a low-level program language. It will be programed for an ARM Cortex-M3 microcontroller. This microcontroller makes it possible to run in different energymodes and that is something that could be useful for this exercise.

2 Background and Theory

The EFM32GG is a microcontroller with the ARM Cortex-M3 processor. Its focus is on low power applications. The microcontroller consists of many elements like I/O-ports, clock, memory, timers and some other interfaces. It's a 32 bit pipelined RISC processor that supports the ARM Thumb instruction set. In the flash memory you can find the exception vector. This table specifies where in memory different handler and exception are located. The reset handler is located here, and defines what the microcontroller is doing when it starts running. [1] & [3]

There are many I/O-ports on the EFM32GG microcontroller. These ports make it possible to read from buttons and control LEDs or other equipment connected to the microcontroller. The I/O-ports are memory mapped which means that the program reads and writes to different addresses in memory. The GPIO is used for pin configuration and the pins can be used as either an input or an output. The GPIO needs to be configured correctly to be used. For example the GPIO_IEN needs to be configured to use interrupts. [2]

The GPIO-interrupts are used to save energy and to make it possible for the CPU to sleep when there is nothing to do. The microcontroller can run in different energy modes and go to sleep when nothin new is happening or after an interrupt. [2]

3 Methodology

This chapter should discuss the details of your implementation for the assignment. Everything related to *how* things were done should go here. Remember to avoid going into too much details, summarize appropriately and try to use figures/charts. Make sure you refer to the figures (such as Figure 3.1) and charts you add in the text. Avoid putting lots of source code here – small code snippets are fine if you want to discuss something specific.

3.1 Testing

Add content in this section that describes how you tested and verified the correctness of your implementation, with respect to the requirements of the assignment.

Figure 3.1: A JPEG image of a galaxy. Use vector graphics instead if you can.

4 Results

There were made three different programs. In the first program couldn't the user control the LEDs, they were programed directly. The second version used a loop and the LEDs could be controlled with a gamepad. There were also made a third version that used interrupts and went to sleep-mode when there was no interrupt. This was the program with lowest power consumption. When nothing was pressed the power consumption was about 6,93 uW. With the two other versions the power consumption was a lot higher when nothing was pressed (about 3-4 mW).

When the user pressed a button would the power consumption increase because the LEDs need current to be lit. The current increased when more buttons was pressed. Best results was observed from the program that uses interrupts and sleep-mode. When all the LEDs were active, all programs had the same current and power consumption. (see fig. 4.1)

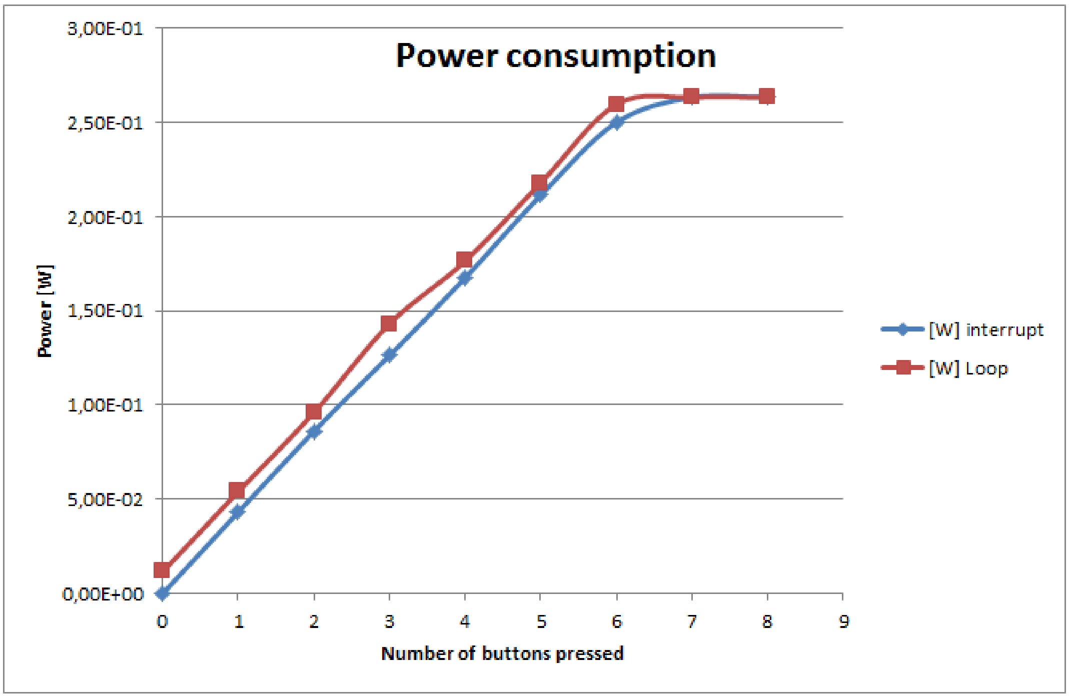

Figure 4.1: Power consumption

5 Conclusion

This chapter should be a look back at the entire report and summarizing the problem, the solution and the obtained results.

5.1 Evaluation of the Assignment

You can include comments about the assignment itself here. While this part is not obligatory and not graded, it is valuable feedback to the course staff that can be used to improve the exercises in the future.

Bibliography

[1] *EFM32GG Giant Gecko Development Kit User's Guide Series.* Silicon Labs, 2013.

[2] *EFM32GG Reference Manual - Giant Gecko Series.* Silicon Labs, 2013.

[3] *Lab Exercises in TDT4258 Energy Efficient Computer Systems.* NTNU, 2014.