Microcontroller lab outline

General course structure:

- 6 sessions of 3h
    - Session 1-4: new content + tasks
    - Session 5-6: project work

The first four sessions each have a presentation (slides in PDF and video) that cover the new content. Each session has a few respective tasks. More information about the project can be found lower in this document.

Overview

Canvas files and folders

Make sure to download all files and folders since you will need them simultaneous at times. In the table below you can find the folder structure and the containing files with a description.

DATASHEETS	Various datasheets
41.T70P015H-LF_Buzzer.pdf	Datasheet of the buzzer (demoboard)
AD5694R_DAC.pdf	Datasheet of the DAC (demoboard)
ArduinoUno.pdf	Arduino UNO schematic
AT328P_microcontroller.pdf	ATmega328p datasheet
FND-157XX00GC_LEDmatrix.pdf	Datasheet of the LED matrix (demoboard)
SME_MicrocontrollerBoard_v2.1.pdf	Schematic of the demoboard
STP08DP05_ShiftRegister.pdf	Datasheet of the shiftregister (demoboard)
ZXM61P03F_transistor.pdf	Datasheet of the screen PMOS (demoboard)
PRESENTATIONS	**Presentations with the lab content**
[see canvas & sections below]	
PRNU	**Pseudo random number generator**
LFSR table.pdf	Table of LFSR seeds
PRNU generator.pdf	PRNG using a linear feedback shift register
PROGRAMMERINFO	**Resources for programming the microcontroller**
AT328P_microcontroller.pdf	ATmega328p datasheet
Atmel-0856-AVR-Instruction-Set-Manual.pdf	Atmel AVR8 instruction set – full manual
Instruction Set Summary.pdf	Atmel AVR8 instruction set – summary
MemoryOverview.xlsx	Excel with the memory and register map
PinoutExpansionHeader.pdf	Pinout of the demoboard expansion connector
ScreenTipsAndTricks.pdf	Tips to make a good screen implementation
SME_MicrocontrollerBoard_v2.1_Schematic.pdf	Schematic of the demoboard
ucontrollers_GetStarted.pdf	Instructions to setup the IDE and toolchain
SOFTWARE	
Installing the Microcontroller board software on your personal computer.pdf	Installation instructions and troubleshooting steps for problem with uploading code to the board.
Portables.zip	Files needed for the toolchain
SoftwareToInstall_Links.txt	Software download link (may be out of date)

Session 1

Content

1 – Introduction

Introduction to the lab and introduction to microcontrollers.

2 – Part 1: Getting started

Getting started with the board and with task 1. After watching follow the instructions in: ***ucontroller_GetStarted.pdf***

ucontroller_GetStarted.pdf: This document contains all the necessary info to get the IDE and toolchain setup. Follow **ALL** instructions well. Skipping over any might result in things not working. If you want to setup the toolchain on your own PC follow the instructions in: "***Installing the Microcontroller board software on your personal computer.pdf"*** This also contains troubleshooting steps which are also applicable to **ucontroller_GetStarted.pdf**

3 – Background 1

Additional information about the microcontroller and programming it in assembly. This presentation contains information about registers, operations, compiler directives, ...

Tasks

Task 1

Turn on the upper led when the button is pressed.

Task 2

Turn on the lower led when the switch is in the high position.

Task 3

Make the led blink visibly.

Note: no need to use a timer yet. Making a led blink using delays is ok.

Task 4

Make the buzzer sound audibly.

Note: no need to use a timer yet.

Session 2

Start with presentation 4 – 7 before starting the tasks, each video contains vital content that needs to be understood before the task can be made.

Content

4 – Part 2: Memory

Presenting the basics about addressing the different parts of the memory. Understanding the memory and addressing are essential in going forward.

Make sure to download: *MemoryOverview.xlsx* this is the actual overview of the memory structure. It will be crucial to check which register is in which memory region.

5 – Part 2: Timers

Explaining the basic operation of the timers of the ATmega328p and how to configure.

6 – Part 2: Interrupts

Presenting the interrupt service routine and how to enable global and time overflow interrupts.

7 – Part 2: Timer interrupts

Combining timer and overflow interrupt to optain a desired freqency, how to handle multible interrupts.

8 – Background 2: Program flow

Additional aspects that are important in understanding how the program flow of a microcontroller works. It goes into more detail of the differences between jumps, calls and macros. The principle behind branching is also explained.

9 – Part 3: Simulator

This part is separate from the session content, but at this point in the lab it might be necessary to start verifying code using the simulator to better understand the principles of operation as well as for debugging. This presentation explains how to use the simulator.

Tasks

Task 5

Make the buzzer sound at 440Hz when the button of the joystick is pressed. Use a timer and an interrupt to achieve this.

Task 6

Make the buzzer sound at 440Hz and 880Hz based on the state of the switch. Use a timer and an interrupt to achieve this.

Session 3

Content

10 – Part 4: Keyboard

Explaining how to readout the keyboard using row scanning in a 4-steps method, as well as a more compact 2-steps method.

Tasks

Task 7

Implement one method to readout the keyboard, write code such that the buttons do the following: Button 7: Two leds on; Button 8: Bottom led on; Button 4: Top Led on; All other buttons: Buzzer on; No buttons pressed: Leds and buzzer off

Session 4

Content

11 – Part 5: Screen

Explaining the basics for the screen.

12 - Part 5b: Screen methods

Presenting the idea behind the two different approaches described in the screen tips and tricks document.

Tasks

Task 8

Get something on the screen. Shift out something into the shift registers of the screen such that at least one LED is on.

Task 9

Make a checkerboard pattern on the screen. Alternating leds on-of-on-off.

Task 10

Show something on the screen such as a character, a symbol, ...

Project (session 5 & 6)

Choose a project to do. The project can be done alone or with 2 people. If you're working with two, consider that the project needs to be elaborate enough to cover the workload of two people.

When choosing a project check with your TAs to see if the project is feasible and the workload is adequate.

The project needs to conform to the following basic requirements:

- Written in assembly
- Use the screen and keyboard in a sensible way, using a clean implementation.
- Use at least 1 timer and 1 interrupt in a sensible way.

If these basic requirements are not fulfilled no defence of the project will be allowed, resulting in a 0/20 for that session.

The project will need to be defended orally. During this defence we will ask you to explain some things about your code, and we will use your implementation as a guide to gauge your knowledge of the basics.

The goal is not just to have you make a project, but to have you understand the basics as well! Just having written code but not being able to explain what you did and why you did it like that will not earn you a passing grade.

Note that **plagiarism will not be tolerated**. Excessive copying of code from previous years or this year's colleagues falls under this. Do not try this, we have tools to detect plagiarism and we will notice it!

Submission is done trough canvas. Check the appropriate assignments on canvas for the deadline.

The deadline has a 24h grace period. If you submit within 24h after the deadline elapsing the project will still be accepted with a penalty of -2/20 on the final grade of the lab. Anything later than 24h will not be accepted. No exceptions!