HOCHSCHULE
RHEIN-WAAL
Rhine-Waal University
of Applied Sciences

LABORATORY

Microelectronic Control Systems

EXPERIMENT:

Retake Microcontroller Programming

Please read the whole document before starting the experiment. This document contains 7 preparation jobs. It is mandatory to make all preparation jobs in written form! Without a written document that includes all preparation jobs it is not possible to pass this experiment.

1 The Lab

In the lab are 20 workplaces with a personal computer and a MyAVR board. The MyAVR board is connected and powered via USB (Universal Serial Bus). The microcontroller used in this lab is a Atmel Mega88pa. To get more information and to download the datasheet please visit http://www.atmel.com/images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf

1.1 MyAVR board

The MyAVR board has the basic input and output devices:

- LEDs
- Keys
- Potentiometer
- Sounder
- LCD display

The ports B, C, and D are used to connect to I/O (Input/Output) devices using wires (example shown in Figure 1).

1.2 Software

The used software is complete free and open source. You can download these software using the links provided below:

- Operating System: Ubuntu http://www.ubuntu.com
- Editor: Gedit, VI, VIM, Geany (use what you like)
- Compiler: avr-gcc http://gcc.gnu.org
- Programmer: avrdude http://www.nongnu.org/avrdude

To build a program and flash it to a microcontroller it is possible to execute every manually. However, it is very time consuming. A more convenient way to do this process is using makefiles.

Prepare 1: What are the necessary steps to execute a C program with a microcontroller? This is, what the make tool normaly does.

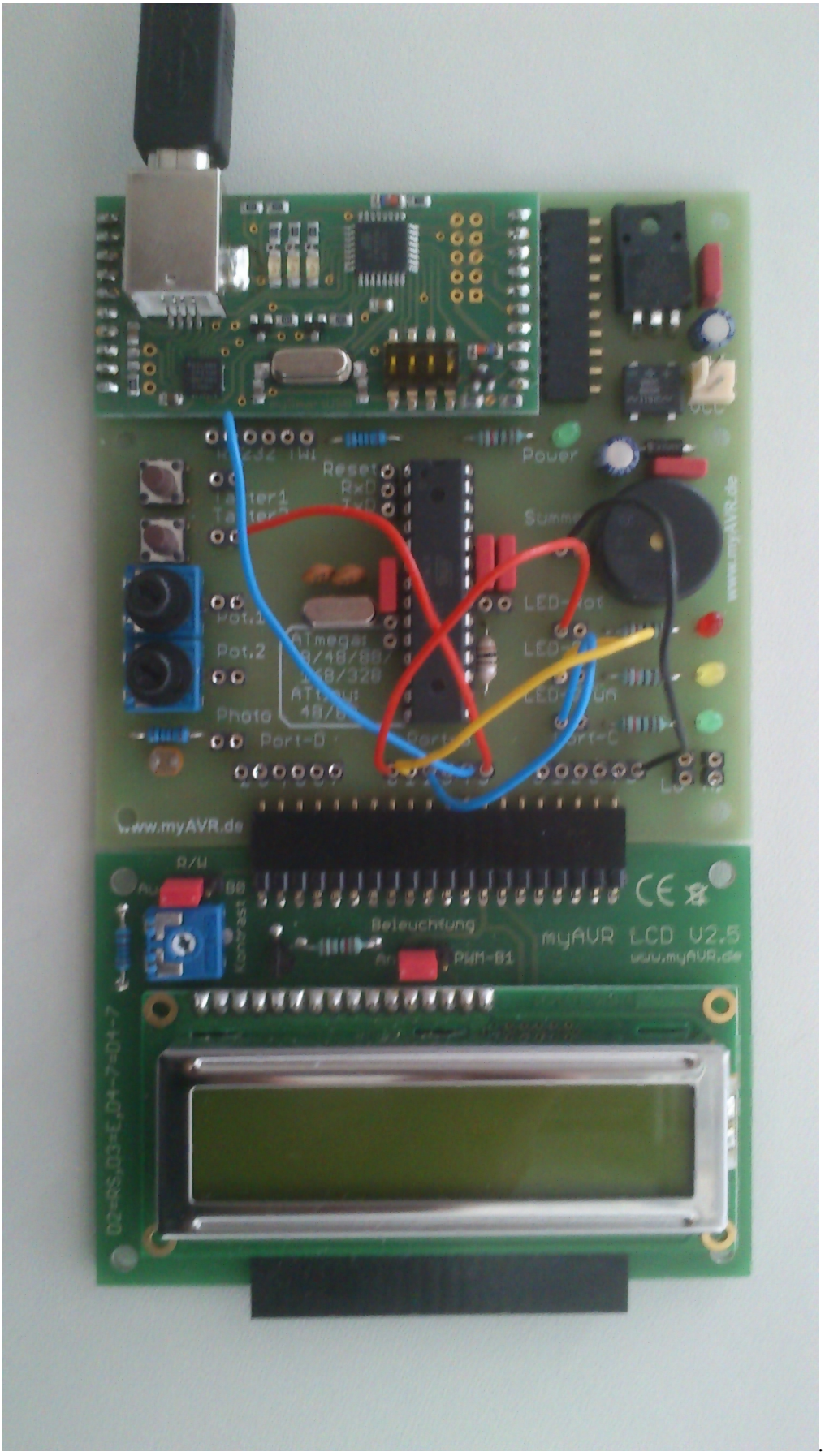

Figure 1: My AVR board with wires

2 Tasks

To retake the microcontroller lab here are some tasks that contain all the important items. Working on this tasks makes only sense, when the lab microcontroller has been done already.

The first step is to compile and flash a pre written program. All files you need are in the directory Templates/Retake.

Open a terminal (STRG + ALT + T) and enter *cd Templates/Retake*.

Prepare 2: Review the basic commands for a Linux shell (cd, vi, make, man, rm, date).

To complete the first step, enter *make flash1*. The LCD display should now show the text: "first step ok".

2.1 Port I/O

Lets use the digital I/O. Connect:

- Key 1 to PB1
- Key 2 to PB0
- The red LED to PC0

With this configuration, flash program two (**make flash2**). With the two buttons you can now enable or disable the LED.

Read and understand the simple source files:

- two.c
- init.c

The following files are for your code:

- three.c: Task in 2.1
- four.c: Task in 2.2
- five.c: Task in 2.3
- six.c: Task in 2.4
- seven.c: Task in 2.5
- eight.c: Task in 2.6

Add more wires and edit the file three.c to perform these functions:

- Key 1 enables the red LED
- Key 2 enables the green LED
- When both Keys are pressed, the red and green LED are disabled
- The yellow LED is on, when any key is pressed, otherwise off

You will get a problem. It is not possible to release both buttons at the same time. So add a line to wait until both buttons are released.

2.2 Simple delay

Sometimes it is necessary to wait a short time. A simple solution is the delay function. Include the program library <util/delay.h> and use the function **_delay_ms(uint8_t time);**.

Write a program that can enable and disable all LEDs (at the same time) with only one key. You will need a short delay for debouncing.

Prepare 3: What is debouncing?

2.3 Analog input

Connect a potentiometer to one of the analog inputs from the microcontroller. Read the value from the analog input and show it on the LCD display. To get help how the analog to digital converter works, read the datasheet.

Hint: The LCD display can only show strings. To show a number on the screen, you need to generate a string that contains this number. You can use sprintf.

Prepare 4: What is the maximum value for a 10 bit analog to digital converter?

Prepare 5: How to use sprintf?

2.4 Timer / Counter

The Atmel Mega88 has three Timer/Counter. They can be used for:

- Time measurement
- Cyclical executions (interrupt)
- PWM generation (can be done by the hardware)
- Counting

Prepare 6: List all timers of the AtMega88 with its maximum value

Write a program to flash one LED with 1 Hz with a Timer/Counter. Find out the best Timer/Counter and prescaler settings to do this. The CPU speed is 8 MHz.

2.5 Interrupts

Interrupts are important and often used. Expand the flashing LED program, that a second LED flashes faster and the third LED is controlled with the buttons. Use only interrupts to realize this. Please disconnect the LCD display and remove all LCD functions to use the external interrupts.

Prepare 7: What is the modifier volatile in C?

2.6 I^2C Bus

The I^2C Bus is a often used data bus for chip to chip communication. The Atmel Mega88 has an internal hardware driver to use this bus. Read the paged 230 to 233 of the datasheet and write a program that checks if an device is connected to the bus.

Connect:

- PC0: Red LED
- PC1: Green LED

Device address:

- EEPROM (All jumpers are set to zero): 0xA0
- DS1307: 0xD0

To test your program, connect and disconnect the TWI add-on modules. The LED with the corresponding color should only be on, if the TWI device is connected:

- Red LED: EEPROM
- Green LED: DS1307