Graphical user interface

Description automatically generated with low confidence

**Laborbericht 1**

**I/O Ports**

**Ausarbeitung Gruppe 20**

Laborbericht 1 eingereicht von

Kühne, Sebastian

Matrikelnummer 2604332

Laborgruppe 20

im Rahmen der Vorlesung *INF3 S22*

im Studiengang Media Systems

am Department Medientechnik der Fakultät DMI

an der Hochschule für Angewandte Wissenschaften Hamburg

Lehrende\*r: Prof. Dr. Tessa Taefi

Eingereicht am: 18.04.2022

**Zusammenfassung**

Im Rahmen der ersten Labor-Übung des Kurses „Informatik 3 und Elektronik“ soll ein Atmel „Atmega 328P“ Board zusammen mit einem 10-LED Shield über selbstentwickelten C Code in Betrieb genommen werden.  
Diese wird im Vorhinein mithilfe von Schaltplänen und UML Aktivitätsdiagrammen geplant und in der Entwicklungsumgebung „Microchip Studio“ getestet, um im Anschluss entsprechende Erkenntnisse wissenschaftlich zu dokumentieren.

**Inhaltsverzeichnis**

[Abkürzungsverzeichnis III](#_Toc101277539)

[Abbildungsverzeichnis IV](#_Toc101277540)

[Formelverzeichnis V](#_Toc101277541)

[1 Labor 1 - I/O PORTS 1](#_Toc101277542)

[1.1 Ziele der Laboraufgaben 1](#_Toc101277543)

[1.2 Anforderungsanalyse 1](#_Toc101277544)

[2 Laborvorbereitung und Methodik 2](#_Toc101277545)

[3 Dokumentation der Laborarbeit 3](#_Toc101277546)

[3.1 Planung der Laboraufgaben 3](#_Toc101277547)

[3.1.1 Xplained Mini Board 3](#_Toc101277548)

[3.1.2 Shield und I/O Ports 4](#_Toc101277549)

[3.1.3 Interrupts 7](#_Toc101277550)

[3.2 Implementation und Testung 8](#_Toc101277551)

[3.2.1 Laden von selbstentwickelter Software auf das Xplained Mini Board 8](#_Toc101277552)

[3.2.2 Inbetriebnahme eines selbstentwickelten LED Shields 9](#_Toc101277553)

[3.2.3 Nutzung von Interrupts auf dem Xplained Mini Board 9](#_Toc101277554)

[4 Diskussion 10](#_Toc101277555)

[4.1 Verwendung des Boards unter Microchip Studio 10](#_Toc101277556)

[4.2 Firmware auf dem Board im Vergleich zu SimulIDE 10](#_Toc101277557)

[5 Fazit 11](#_Toc101277558)

[Literaturverzeichnis 12](#_Toc101277559)

[Anhang 13](#_Toc101277560)

[Eigenständigkeitserklärung 14](#_Toc101277561)

Abkürzungsverzeichnis

MCS Microchip Studio

**HIER REGISTERNAMEN ERLÄUTERN**

Abbildungsverzeichnis

[Abbildung 1: Schatplan Atmega 328P mit button und LED 3](#_Toc101250293)

[Abbildung 2: Blockschaltbild LEDOnButtonPress 3](#_Toc101250294)

[Abbildung 3: UML Aktivitätsdiagram OnButtonPress 3](#_Toc101250295)

[Abbildung 4: UML Aktivitätsdiagram CountAndBlink 4](#_Toc101250296)

[Abbildung 5: Schaltplan Atmega 328P mit selbstentwickeltem LED Shield 4](#_Toc101250297)

[Abbildung 6: Blockschaltbild Atmega 328P mit Shield 5](#_Toc101250298)

[Abbildung 7: Blockschaltbild CountButtonAndBlink mit Shield 5](#_Toc101250299)

[Abbildung 8: UML Aktivitätsdiagram CountButtonAndBlink 6](#_Toc101250300)

[Abbildung 9: UML Aktivitätsdiagram CountButtonShowBinary 6](#_Toc101250301)

[Abbildung 10: UML Aktivitätsdiagramm ToggleWithPolling 7](#_Toc101250302)

[Abbildung 11: UML Aktivitätsdiagram ToggleLEDsWithInterrupts 7](#_Toc101250303)

[Abbildung 12:LEDOnButtonPress - main.c 8](#_Toc101250304)

[Abbildung 13: Ausschnitt CountButtonAndBlink - main.c 9](#_Toc101250305)

Formel Verzeichnis

[1 Ohm's Law 5](#_Toc101241732)

# Labor 1 - I/O PORTS

Ausbreitung Fragestellung

* INFC und Elektronik LABOR
* Abnahme durch praktische Ausführung während einer Laboreinheit
* Verständnis Atmel 328P und Ateml Microchip Studio (fortlaufend „MCS“)
* Inbetriebnahme eines selbstentwickelten LED Shields
* Steuerung der IO Ports des Boards
* Verständnis und Nutzung von Interrupts

## Ziele der Laboraufgaben

* Erfolgreiches Ausführen selbstgeschriebener Firmware auf dem Board
* Einbindung und Kontrolle eines selbstentwickelten LED Shields
* Demonstration erlernter Kenntnisse durch verschiedene selbstentwickelte Programme

## Anforderungsanalyse

Die Anforderungsanalyse ist der Kernabschnitt der Konzepterstellung. Hier definieren Sie die konkrete Anforderungen und messen nach Abschluss der Arbeit an diesen Anforderungen, ob sie die Arbeit erfolgreich abgeschlossen haben, d.h. ob die Anforderungen erfüllt werden.

* **A1** Anfroderungen Microchip Studio und Board
  + Testprogramm läuft auf board
  + Beschreibung CountButtonBlink
* **A2** Anforderung LED Shield
  + Berrechnung LED Shield
  + LED OnButtonPress Button 1
  + Erweiterung Reset
  + Dualzahll Darstellung LEDonButtonPress
* **A3** Nutzung Makros
  + Polling Toggle
  + Nutzung Interrups

# Laborvorbereitung und Methodik

Sämtliche Aufgaben wurden unter MS Windows 10 und teilweise über ein druch USB verbundenes Atmel 328P Board ausgeführt, dazu wurden folgende Software und Datensätzen genutzt.

* Entwicklung von Firmware anhand vorher erstellter Ablaufdiagramme;
* Schreiben von Code und Debugging der Registerzustände in Atmel Microchip Studio
* Testen von Firmware und nachbau entsprechender Schaltpläne in SimulIDE
* Verwendung von durch Ateml zu verfuegunf gestellten Header Datein
* Dokumentation von Atmel zum 328P „Xplained mini“ Board
* Vorlesung

Argumentation Pro Contra Microchip Studio und SimulIDe

Atmel Microchip studio erlaubt durch eingebaute Chip Simulation eine Einsicht in Register und HexDump während der Laufzeit und bietet Hardware Debugging

SimulIDE bietet eine Alternative Firmware zu testen sofern kein Chip zur hand ist, weißt aber verschiedene Bugs auf und kann unzuverlässige Ergebnisse liefern

Limitation:

* Limitiertes vorwissen zum Board
* Debugging nur ohne Delay möglich
* Fehelen eines seperaten Debugging Boards
* Verfuegbarkeit von Boards in der Vorbereitung
* Buggs in SimulIDE

# Dokumentation der Laborarbeit

Ablauf Laborübung und Bearbeitung der entsprechenden Aufgaben

## Planung der Laboraufgaben

Planung und Konzept für das Labor -> Generelle Vorbereitung durch Vorlesung und Eigenarbeit

### Xplained Mini Board

Beschreibung der Software LEDOnButtonPress und Aufgabe 1A

Chart, diagram, schematic

Description automatically generated

Abbildung 1: Schatplan Atmega 328P mit button und LED

Chart, treemap chart

Description automatically generated

Abbildung 2: Blockschaltbild LEDOnButtonPress

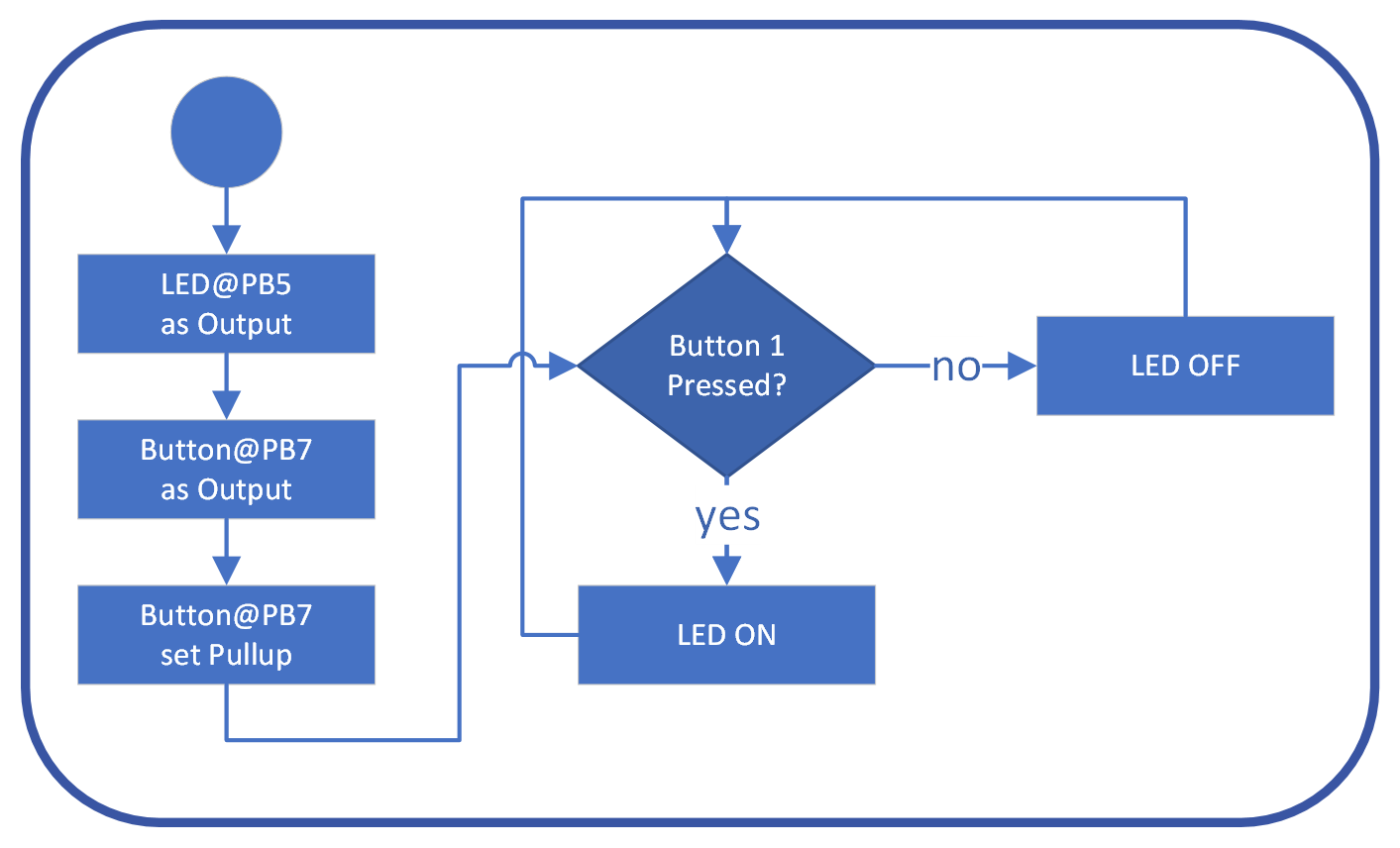


Abbildung 3: UML Aktivitätsdiagram OnButtonPress

Erläuterung CountAndBlink -> Userexperience und planung der software

Verweis auf UML Aktivitäsdiagramm

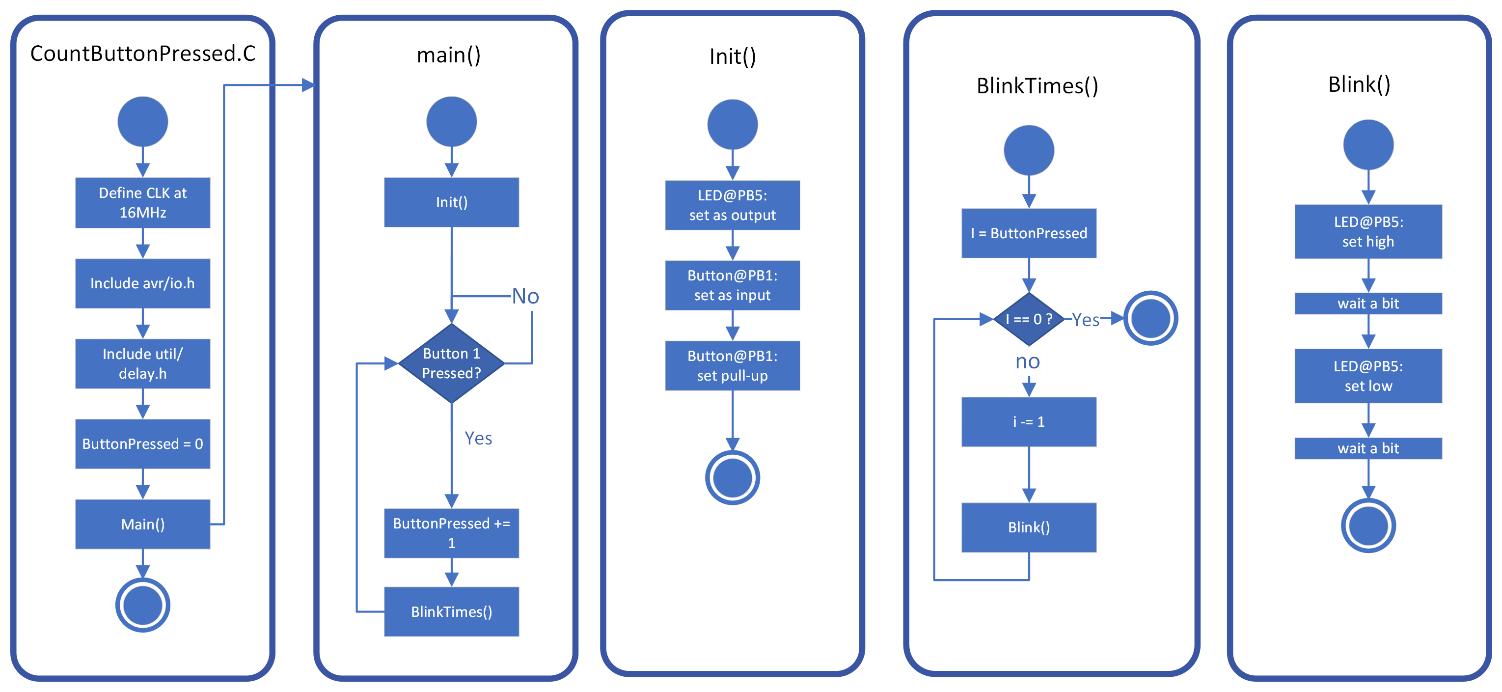


Abbildung 4: UML Aktivitätsdiagram CountAndBlink

text

### Shield und I/O Ports

Wie in Laboraufgabe 2A beschrieben, soll vor der Inbetriebnahme des Shields geprüft werden, ob der Controller genug Strom für das 10 Segment LED-Display des Shields liefert.   
Mithilfe des in Abbildung 5 gezeigten Schaltplanes und dem in Abbildung 6 gezeigten Datenblatt des LED Displays, lassen sich folgende Werte ermitteln:

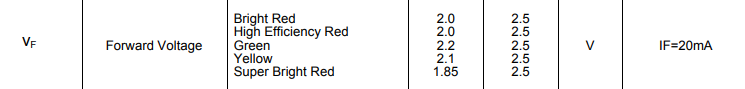


Abbildung : Auszug Knightbrigth DC-10E Datasheet

V = 5 Volt | R = 220 Ohm | UR Green = 5 – 2.2 = 2.8V | UR Bright Red = 5 – 2 = 3V

Ohm's Law

Mit sieben LEDs vom Typ „Green“ á 12mA, und drei vom Typ „Bright Red“ á 13mA lässt sich der Gesamtbedarf des Shields auf ca. 123mA berechnen. Somit überschreitet der Gesamtbedarf des Shields nicht das im ATmega328P angegebene Maximum von 150mA für Ports, und ist mit 20mA pro Output Pin pro LED ebenfalls gedeckt.

Diagram, schematic

Description automatically generated

Abbildung : Schaltplan Atmega 328P mit selbstentwickeltem LED Shield

Nach dem Verbinden des Shields soll nun das bereits

Neues Blockdiagramm mit Shield und Verweis auf altes UML Aktivitätsdiagramm

Chart, treemap chart

Description automatically generated

Abbildung 7: Blockschaltbild Atmega 328P mit Shield



Abbildung : Board mit aufgestecktem Shield

Ergebnisse 2C

Beschreibung CountButtonAndBlink neue Funktion: RESET!

Neues Blockdiagramm 1D -> 2 Buttons mit Shield

A screenshot of a computer

Description automatically generated with medium confidence

Abbildung 9: Blockschaltbild CountButtonAndBlink mit Shield

Erläuterung und Planung des Resets, verweis auf UML Aktivitätsdiagramm

Main verfügt nun über eine weitere „wenn dann“ Verzweigung

A screenshot of a computer

Description automatically generated with low confidence

Abbildung 10: UML Aktivitätsdiagram CountButtonAndBlink

Planung von CountButtonShowBinary -> Verweis auf UML

Entprellen des Buttons!

Konzept schreiben des Counters in Led-Register zur anzeige

A picture containing text, iPod

Description automatically generated

Abbildung 11: UML Aktivitätsdiagram CountButtonShowBinary

### Interrupts

Beschreibung der LED Register und verweis auf UML

Polling Planung

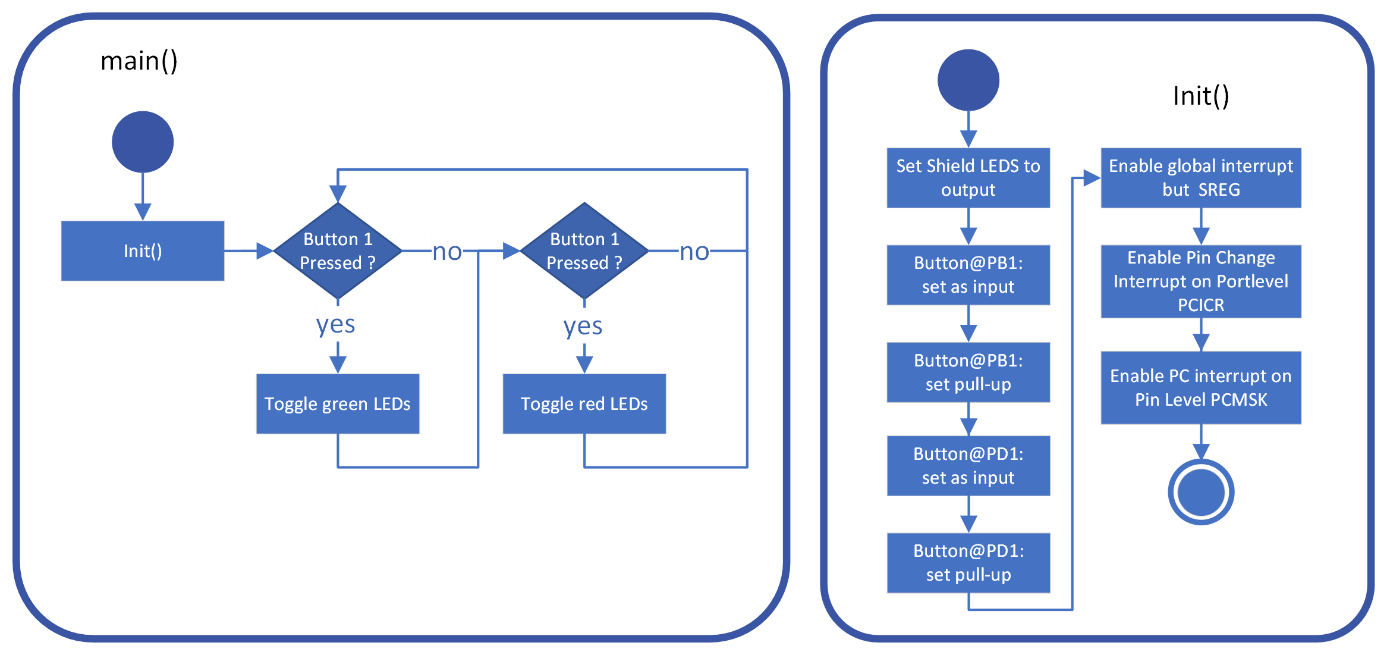


Abbildung 12: UML Aktivitätsdiagramm ToggleWithPolling

Einsatz Interrupts

A screenshot of a computer

Description automatically generated with low confidence

Abbildung 13: UML Aktivitätsdiagram ToggleLEDsWithInterrupts

Erläuterung Implementation mithilfe von Datenblatt?

## Implementation und Testung

Hier reden wir den Laborablauf

### Laden von selbstentwickelter Software auf das Xplained Mini Board

#include <avr/io.h>

#define LED\_ON PORTB |= (1<<5)

#define LED\_OFF PORTB &= ~(1<<5)

#define LED\_TOGGLE PINB |= (1<<5)

#define BUTTON\_PRESS (!(PINB & (1<<7)))

void init(void){

    // LED

    DDRB |= (1<<5); // Configure PB5 as Output

    LED\_ON;

    // Button

    DDRB &= ~(1<<7);//Configure PB1 as Input

    PORTB |= 1<<7;  //Enable Internal Pull-Up at PB1

}

int main(void){

    init();

    while (1){

        if(BUTTON\_PRESS)

            LED\_ON;

        else

            LED\_OFF;

    }

}

Abbildung 14: LEDOnButtonPress - main.c

void blink(unsigned n){

    for(unsigned i = 0; i <= n; i++){

        LED\_ON;

        \_delay\_ms(1000);

        LED\_OFF;

        \_delay\_ms(1000);

    }

}

int main(void){

    init();

    while (1){

        if(BUTTON\_PRESS){

            blink(buttonPresses ++);

        }

    }

}

Abbildung 15: Ausschnitt CountButtonAndBlink - main.c

### Inbetriebnahme eines selbstentwickelten LED Shields

Asfdgd

Anpassung der Ports in Software

Setzen von ganzen Registern

Codesnippet

### Nutzung von Interrupts auf dem Xplained Mini Board

Asdfsadf

CodeSnippet Makros

# Diskussion

Rückbezug zur Vorlesung

Waren die Ergebnisse wie erwartet?

**Beobachtungen zu 2C -> LED zeigt keine Reaktion**

Was war überraschend?

**Beobachtung 2B -> Taster schaltet manchmal zu weit**

**Diskussion zur Durchführung von 2E -> Counter kann einfach in Register gesetzt werde wessen Bits anschließen einfach um eins verschoben werden**

**Anpassung durch Software Prävention nach der Beobachtung von Prellen**

## Verwendung des Boards unter Microchip Studio

Dsfasdgasdg

Nutzung von Hardware Debugging während der Entwicklung

Eingang auf Limitation z.B. Vorwissen

## Firmware auf dem Board im Vergleich zu SimulIDE

asfsdafdsgsdag

Probleme in SimulIDE während des Testings

Eingang auf verfuegbarkeit der Hardware

# Fazit

Was gelernt wurde,

Antworten auf Fragen in 1

Wir haben custom software auf den Controller geladen

Wir haben die I/O Pins zur steuerung eines Shields genutzt

Wir haben dabei Prellen beobachtet

Wir haben den Unterschied zwischen Polling und Interrupts gelernt

Literaturverzeichnis

# 

Anhang

Eigenständigkeitserklärung

Hiermit versichern wir, dass wir das vorliegende Dokument selbständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die wir wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen haben, haben iwir deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

18. April 2020

|  |  |  |
| --- | --- | --- |
| Datum |  | Unterschrift |