



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Laborbericht 1

Shield, GPIO, LEDBank, Tastenprellen

Bericht des ersten Labors eingereicht von

Gödecke, Clara

Matrikelnummer 2579269

im Rahmen der Vorlesung *Informatik und Elektronik*

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: 10.04.2022

Zusammenfassung

Im Rahmen des Kurses „Informatik 3 und Elektronik“ von Prof. Dr. Tessa Taefi galt es im ersten Labor sich mit „I/O Ports“ auseinanderzusetzen. Die in der zweiten Vorlesung behandelte I/O Programmierung sollte in diesem Labor praktisch angewandt werden.

In diesem Labor konnten demzufolge I/O Ports und Programmierung praktisch umgesetzt und angewandt werden. Es wurden Anforderungen analysiert, typische Peripheriemodule eines Mikrocontrollers konfiguriert, Module in Programmiersprache C programmiert, wodurch wir nun in der Lage sind das Atmel ATmega328P Xplained Mini Board besser zu verstehen und erste Programme debuggen konnten.

Im Folgenden ist die Dokumentation dieses Labors und die Umsetzung der genannten Inhalte zu finden. Der Code zu den Programmen wurde auf Emil hochgeladen.

Inhaltsverzeichnis

Abbildungsverzeichnis	III
1. Einleitung.....	I
2. ATmega 328P und SimulIDE.....	I
3. Ergebnisse.....	II
3.1. Aufgabe 1 - Xplained Mini Board.....	II
3.2. Aufgabe 2 - Shield.....	III
3.3. Aufgabe 3 - Interrupts.....	VII
4. Fazit.....	X
Eigenständigkeitserklärung.....	XI

Abbildungsverzeichnis

Abbildung 1: Blockdiagramm CountButtonAndBlink	II
Abbildung 2: Schaltbild CountButtonAndBlink	II
Abbildung 3: Aktivitätsdiagramme CountButtonAndBlink	III
Abbildung 4: Blockdiagramm LEDOnButtonPress mit Shield	IV
Abbildung 5: Schaltbild LEDOnButtonPress mit Shield	IV
Abbildung 6: Aktivitätsdiagramme LEDOnButtonPress mit Shield	IV
Abbildung 7 und 8: Blockdiagramme LEDOnButtonPress mit Erweiterung um Button 2	V
Abbildung 9: Aktivitätsdiagramme LEDOnButtonPress mit Erweiterung um Button 2	VI
Abbildung 10: Blockdiagramme LEDOnButtonPress mit Erweiterung um Button 2 (dual)	VI
Abbildung 11: LEDOnButtonPress mit Erweiterung um Button 2 (dual)	VII
Abbildung 12: Blockdiagramme Interrupts	VIII
Abbildung 13: Schaltbilder Interrupts	VIII
Abbildung 14: Aktivitätsdiagramme Polling	IX
Abbildung 15: Aktivitätsdiagramme Interrupts	IX

1. Einleitung

Gegenstand dieses Laborberichtes ist es erste Schritte mit Atmel ATmega328P Xplained Mini Board nachvollziehen und bearbeiten zu können. Das Atmel ATmega328P Xplained Mini Board ist eine Atmel-Entwicklungsplatine und steht im Fokus dieser und der nächsten beiden Laborarbeiten und zeichnet sich als Schwerpunkt der Inhalte des Kurses „Informatik 3 und Elektronik“ aus. Um den ATmega 328P besser kennenlernen zu können und mit diesem arbeiten zu können, galt es sich in diesem Labor damit auseinanderzusetzen.

Das Ziel des Labors war es die verwendete Hardware in einem Blockdiagramm darstellen zu können, ein für dieses Labor entwickelte Shield mithilfe des Schaltplans in Betrieb zu nehmen und die Software in einem UML-Aktivitätsdiagramm zu planen und zu dokumentieren. Darüber hinaus sollten Programme für das Atmel ATmega328P Xplained Mini Board mithilfe des Microchip Studios geschrieben und debugged, sowie die I/O Ports zur Ein- und Ausgabe konfiguriert werden. Auch ein selbstentwickeltes Shield sollte erfolgreich in Betrieb genommen werden, ebenso die Nutzung von Interrupts angewandt werden.

Da auf der Basis dieser Labors in den nächsten Laboren weiter mit dem ATmega 328P Microcontroller gearbeitet wird, war es wichtig die oben genannten Schritte gut nachvollziehen und umsetzen zu können. Dies gestaltet sich neben den eigentlichen Aufgaben als Ziel dieses Labors.

Die Anforderung an uns selbst war es dementsprechend die Aufgaben genau und detailliert zu bearbeiten. Auch Analyse, Design, Implementierung und Ergebnisse dachgerecht zu dokumentieren war Teil dessen und

2. ATmega 328P und SimulIDE

Um die Anforderungen dieses Labors umsetzen zu können, sollte die Entwicklungsumgebung Microchip Studio, sowie die Simulationsumgebung SimulIDE installiert werden. So konnten wir nicht nur Programme selber schreiben, sondern diese direkt auch auf den ATmega 328P simulieren und überprüfen, ob sie funktionieren. Wir haben die Umgebungen an einem Windows Desktop-Computer installiert und genutzt, da dies den Vorteil hatte auch von zuhause aus arbeiten zu können und der ATmega 328P in der Simulationsumgebung genutzt werden konnte und dieser nicht gekauft werden musste.

Und die ersten Programme mit dem Atmel ATmega328P Xplained Mini Board umsetzen zu könne, wurde die Programmiersprache C verwendet. C hat den Vorteil, dass Programme geschrieben werden können, die schnell in der Ausführung sind und wenig Ressourcen benötigen. Zudem ist der Sprachumfang sehr klein und die mit C geschriebenen Programme können einfach auf andere Systeme sortiert werden.

3. Ergebnisse

3.1. Aufgabe 1 - Xplained Mini Board

Das als Hausaufgabe bereits vorbereitete Programm „LEDOnButtonPress“ sollte zuerst zum Laufen gebracht und daraufhin ein neues Programm „CountButtonAndBlink“ geschrieben werden. Dieses soll fortlaufend die Anzahl der „Button Presses“ zählen und die LEDs dementsprechend oft blinken lassen. Beide Programme konnten mithilfe von Microchip Studio und des Atmel ATmega328P Xplained Mini Board zum Laufen gebracht werden.

Abbildung 1 zeigt das Blockdiagramm zu dieser Aufgabe und beschreibt den Zusammenhang zwischen Sensor, Prozessor und Aktor.

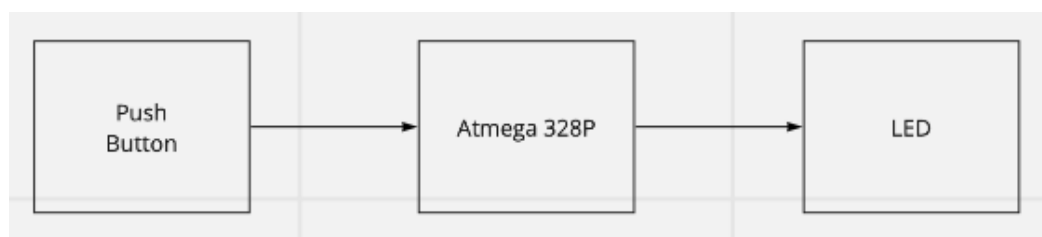


Abbildung 1: Blockdiagramm CountButtonAndBlink

Quelle: Eigene Darstellung

Abbildung 2 zeigt das Schaltbild zu CountButtonAndBlink.

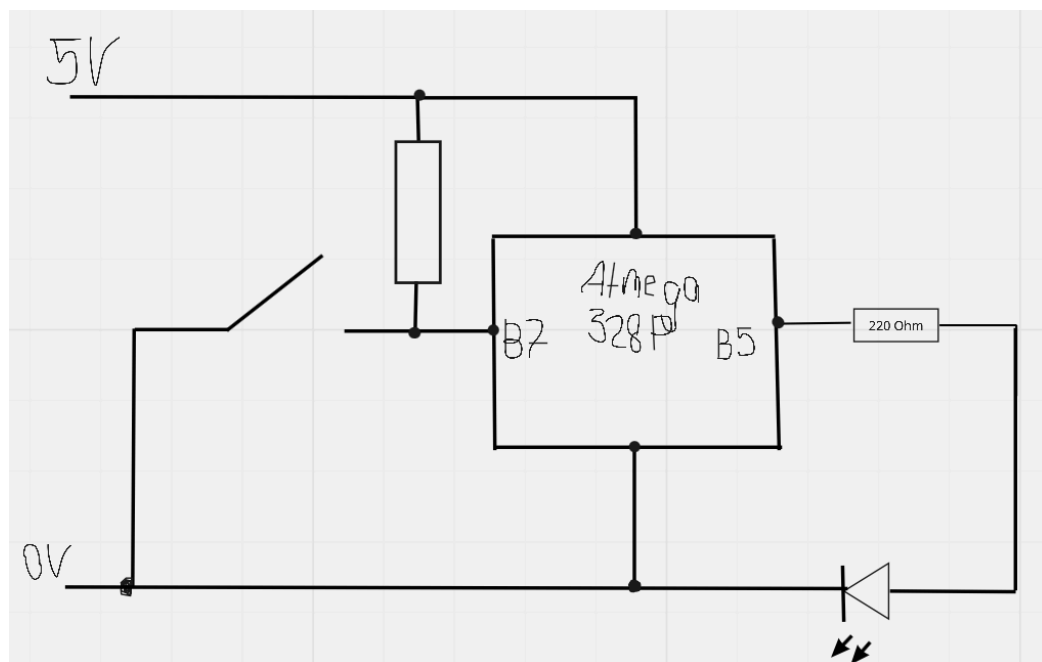


Abbildung 2: Schaltbild CountButtonAndBlink

Quelle: Eigene Darstellung

Abbildung 3 zeigt die Aktivitätsdiagramme zu CountButtonAndBlink. Diese zeigen eine übersichtliche Darstellung von den Abläufen und verdeutlichen, in welcher Reihenfolge die Aktivitäten ausgeführt werden.

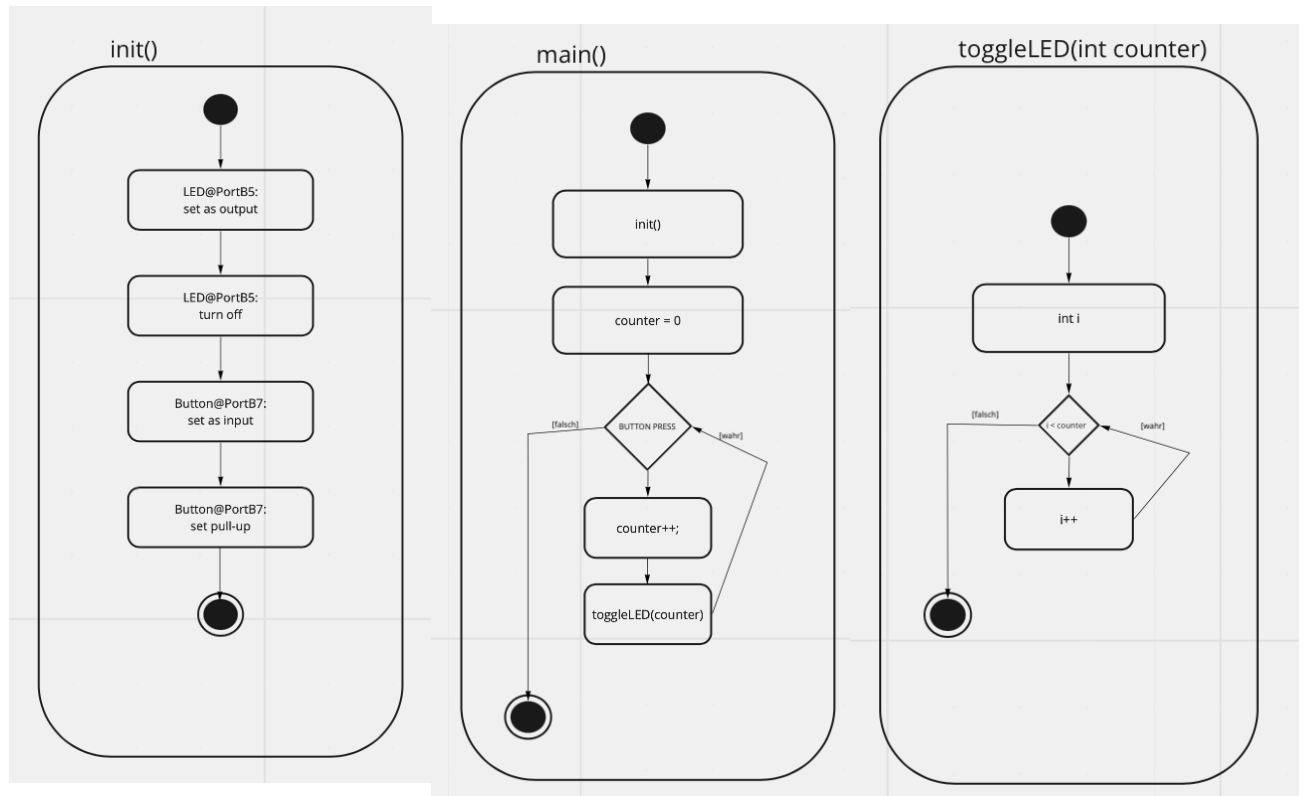


Abbildung 3: Aktivitätsdiagramme
CountButtonAndBlink

Quelle: Eigene Darstellung

3.2. Aufgabe 2 - Shield

Daraufhin sollte ein selbstentwickeltes Shield auf das Xplained Board gesteckt werden. Zuerst galt es zu errechnen, ob der Strom, den die Ports zur Verfügung stellen, für alle LEDs ausreichend ist. Die Rechnung wird im Folgenden einmal dargestellt.

U = Spannung, R = Widerstand, i = Strom

Gegeben sind:

$U = 5 \text{ Volt (V)}$

$R = 220 \text{ Ohm } (\Omega)$

$U_{LED} = 2.3 \text{ V}$ (optimale Voltzahl, siehe Formblatt)

Formeln: $i = U/R$, $U = R \cdot i$, $R = U/i$

Rechnung:

1 LED: $(5V - 2.3V) / 220\Omega = 0.01227 A = 12.27 mA$

10 LED: $10 * 12.27mA = 122.7mA \approx 120mA$

Die Antwort lautet, dass im Atmega 328P 150 mA möglich sind, dementsprechend ist der Strom, den die Ports zur Verfügung stellen, auch für 10 LEDs ausreichend.

Aufgabe 2b) beschäftigt sich mit dem Programm LEDOnButtonPress mit dem selbstentwickelten Shield. Hierbei soll Button 1 so angesteuert werden, dass die erste grüne LED blinkt.

Abbildung 4 veranschaulicht dies in einem Blockdiagramm.

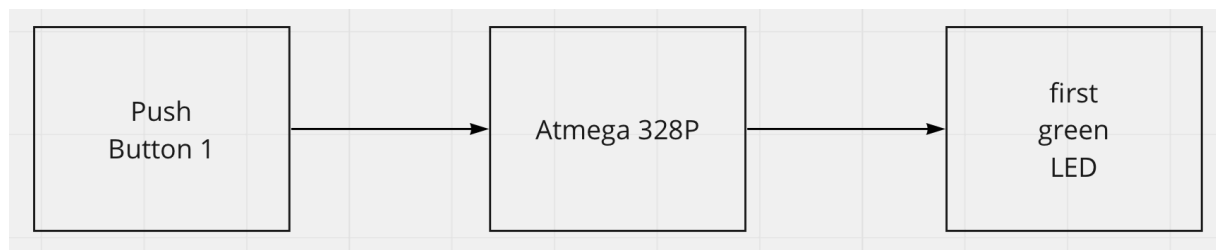


Abbildung 4: Blockdiagramm LEDOnButtonPress mit Shield

Quelle: Eigene Darstellung

Abbildung 5 zeigt das Schaltbild zu LEDOnButtonPress mit Shield.

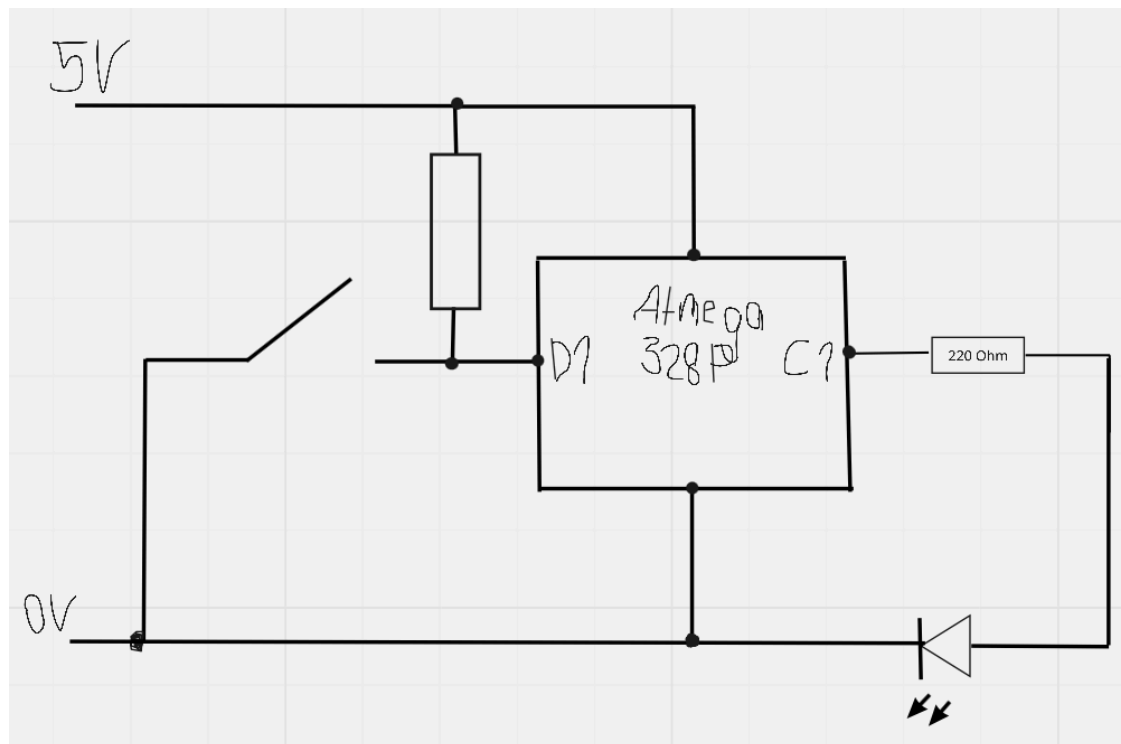


Abbildung 5: Schaltbild LEDOnButtonPress mit Shield

Quelle: Eigene Darstellung

Abbildung 6 stellt LEDOnButtonPress in Aktivitätsdiagrammen dar.

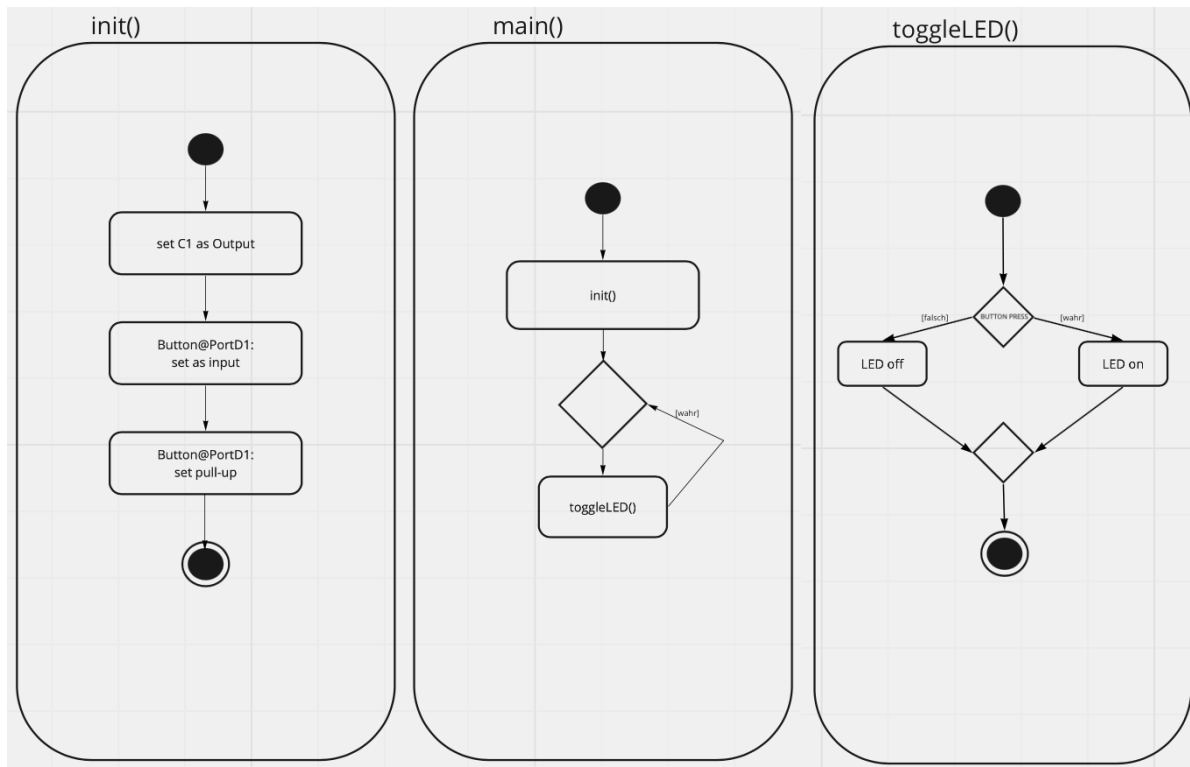


Abbildung 6: Aktivitätsdiagramme LEDOnButtonPress mit Shield

Quelle: Eigene Darstellung

Hier sollte zudem ausprobiert werden, was passiert, wenn der Pullup Widerstand des Push-Buttons entfernt wird. Bei uns ging die LED zwar an, jedoch nicht mehr aus. Ohne den Pullup Widerstand werden die 5 V als Standard genommen und nicht Ground (0 V), weshalb die LED nicht mehr ausgeht.

Dieses Programm sollte zudem in den letzten zwei Teilaufgaben erweitert bzw. abgeändert werden. In Aufgabe 2d) sollte mithilfe des Buttons 2 die Anzahl der Blinks zurückgesetzt werden.

Abbildung 7 und 8 zeigt die zwei Blockdiagramme dazu.

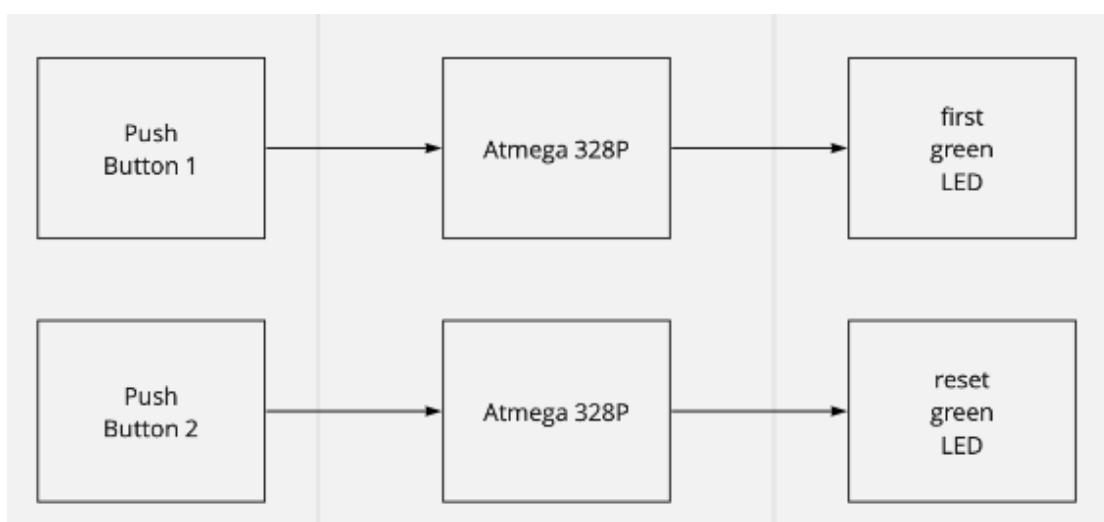


Abbildung 7 und 8: Blockdiagramme LEDOnButtonPress mit Erweiterung um Button 2

Quelle: Eigene Darstellung

Abbildung 9 veranschaulicht das Programm in mehreren Aktivitätsdiagrammen.

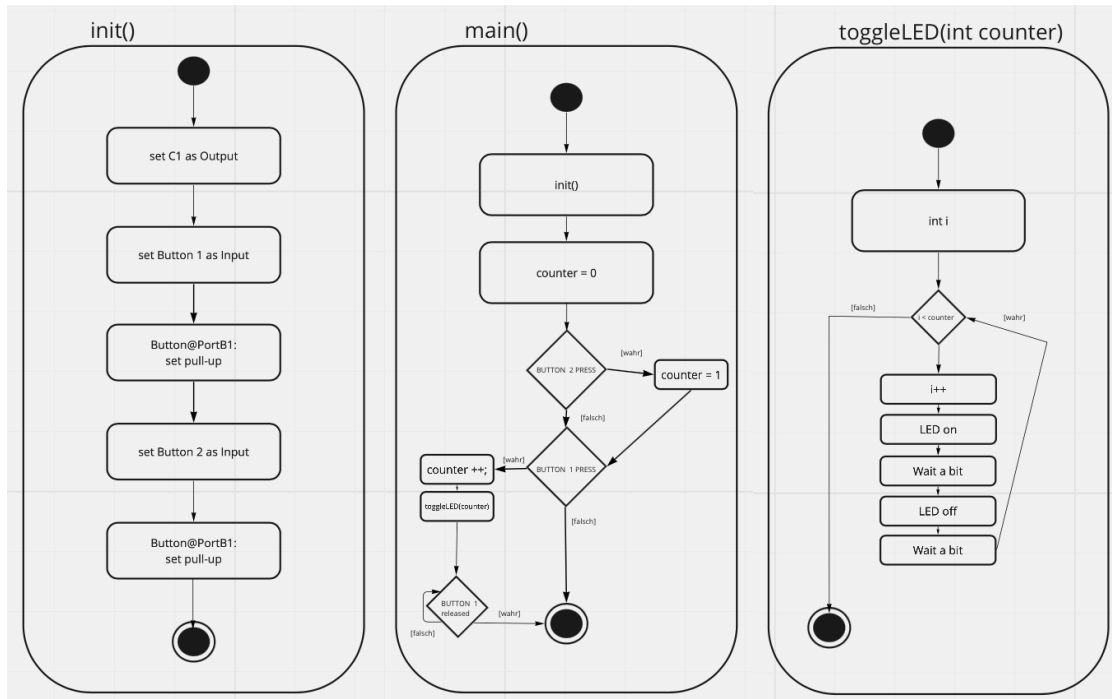


Abbildung 9: Aktivitätsdiagramme LEDOnButtonPress mit Erweiterung um Button 2.

Quelle: Eigene Darstellung

In Aufgabe 2e) galt es das Programm so abzuändern, dass anstatt zu blinken, die Anzahl der gezählten Button presses als vorzeichenlose Dualzahl ausgegeben wird. Dabei sollte die maximal darstellbare Zahl 15 sein, bei einem Überlauf sollte wieder von 0 anfangen werden zu zählen und um einen Reset zu ermöglichen sollte Button 2 gedrückt werden.

Abbildung 10 zeigt das Blockdiagramm zu diesem Programm.

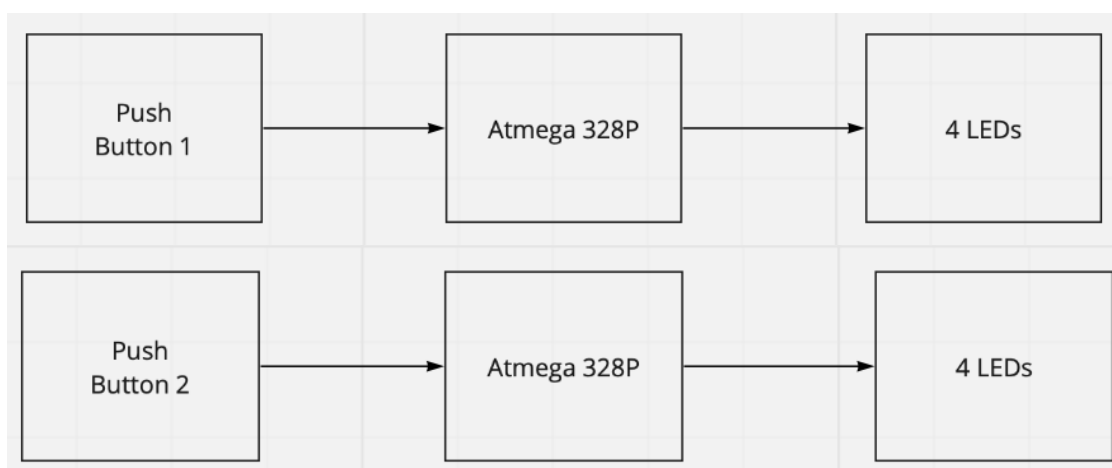


Abbildung 10: Blockdiagramme LEDOnButtonPress mit Erweiterung um Button 2 (dual)

Quelle: Eigene Darstellung

Abbildung 11 präsentiert die passenden Aktivitätsdiagramme zu LEDOnButtonPress mit Erweiterung um Button 2 (dual).

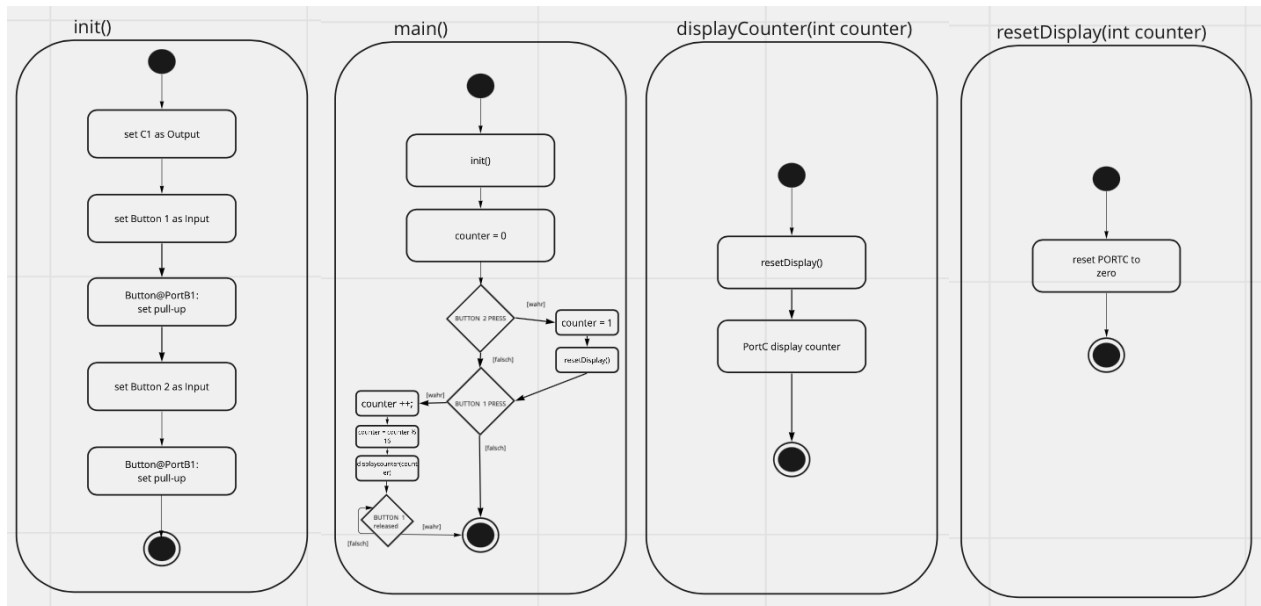


Abbildung 11: LEDOnButtonPress mit Erweiterung um Button 2 (dual)

Quelle: Eigene Darstellung

3.3. Aufgabe 3 - Interrupts

In der dritten und letzten Aufgabe sollte ein Programm entwickelt werden, wo besonders der Fokus auf den in Vorlesung 3 behandelten „Interrupts“ und Polling lag. Solange Button 1 gedrückt wird, sollten alle grünen LEDs angeschaltet sein, während der Button 2 gedrückt wird, die roten LEDs angeschaltet werden sollte. Auch dieses Programm konnten den Anforderungen gemäß umgesetzt und zum Laufen gebracht werden.

Abbildung 12 zeigt hierbei die Blockdiagramme für Aufgabenteil a) und b).

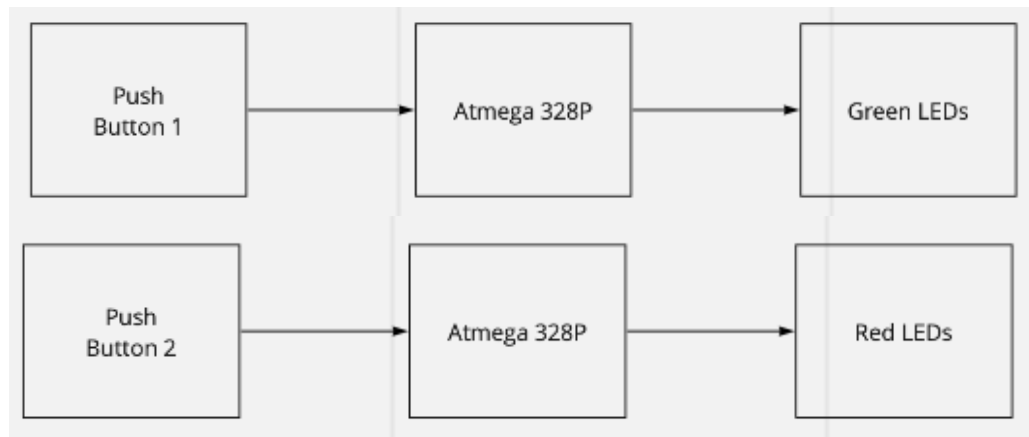


Abbildung 12: Blockdiagramme Interrupts

Quelle: Eigene Darstellung

Abbildung 13 präsentiert das Schaltbild für Aufgabenteil a) und b).

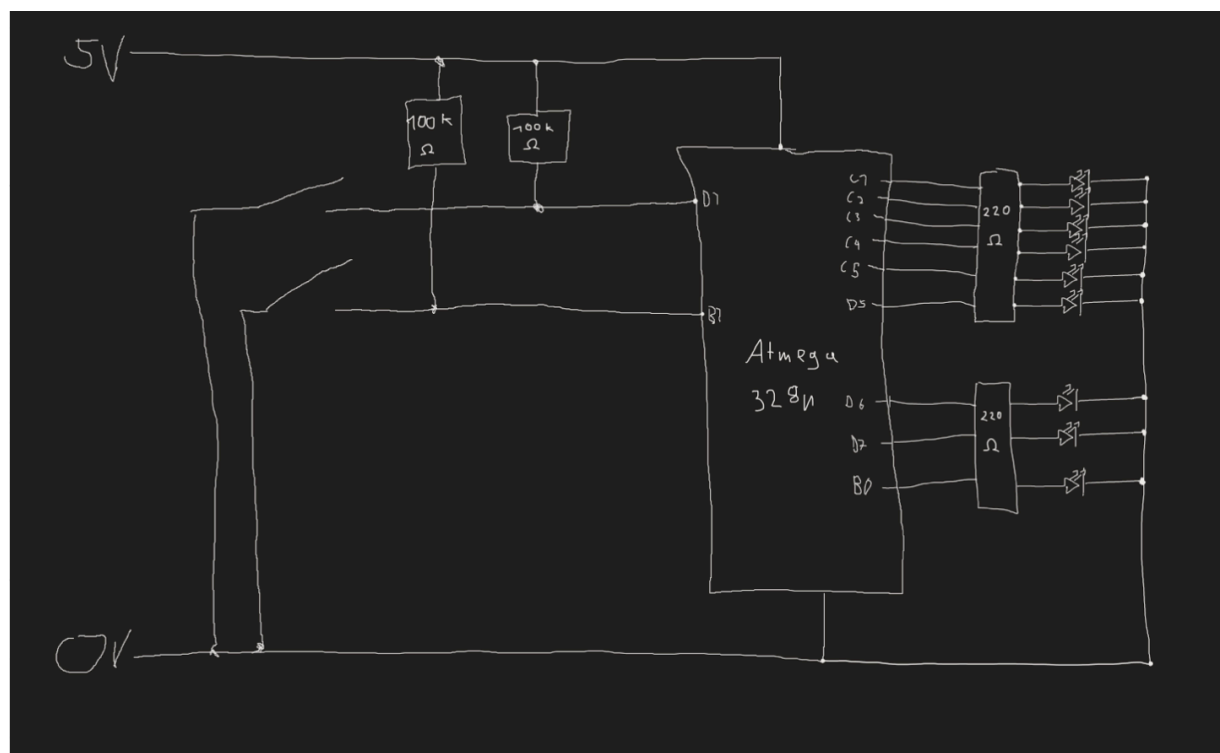


Abbildung 13: Schaltbilder Interrupts

Quelle: Eigene Darstellung

Abbildung 14 stellt die Aktivitätsdiagramme zu Aufgabenteil a) dar.

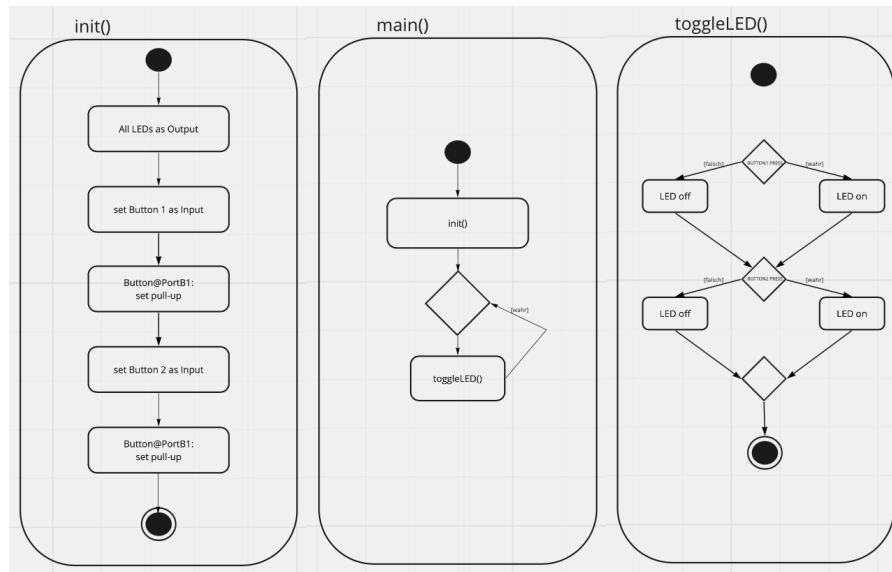


Abbildung 14: Aktivitätsdiagramme Polling

Quelle: Eigene Darstellung

Abbildung 15 stellt die Aktivitätsdiagramme zu Aufgabenteil b) dar.

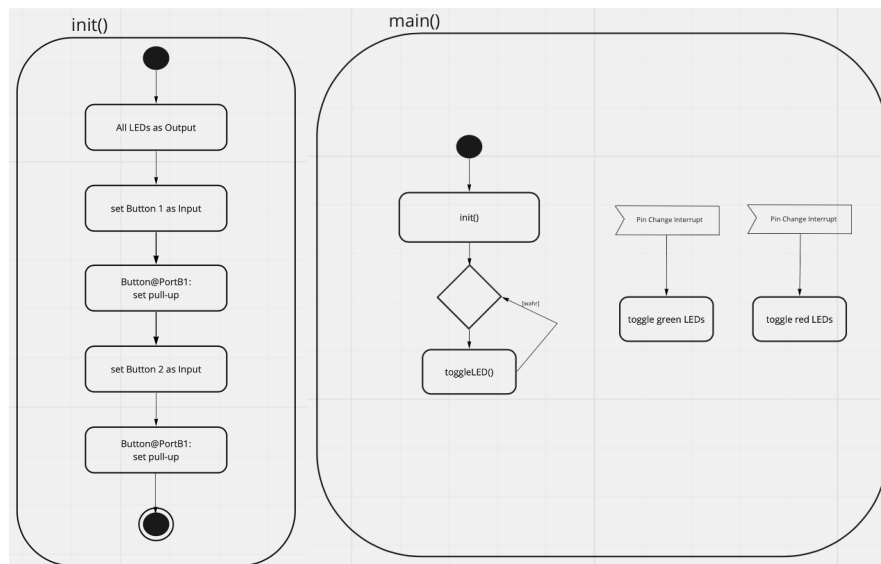


Abbildung 15: Aktivitätsdiagramme Interrupts

Quelle: Eigene Darstellung

4. Fazit

Abschließend lässt sich sagen, dass die geforderten Aufgaben mit mehr oder weniger Schwierigkeiten und Zeitaufwand bearbeitet werden konnten. Uns ist aufgefallen, dass es wirklich wichtig ist sich die Aufgaben ganz genau durchzulesen und in einem Schaltplan zu verwirklichen und diesen anschaulich zu gestalten. Auf diesen kann dann während der Aufgabe immer wieder zurückgegriffen werden.

Auch haben wir an manchen Stellen unerwartet lange gebraucht, wo wir dann genau hinterfragen mussten, wieso genau etwas gerade nicht so funktioniert wie man es erwartet hat. Uns hat besonders die dritte Aufgabe einiges an Zeit gekostet, da leider die LED10RT laut der Tabelle an PD0 gesetzt ist, jedoch eigentlich an PB0 war. Natürlich war das in diesem Fall nicht uns zu verschulden, aber uns ist dabei aufgefallen, wie schnell man sich an Kleinigkeiten aufhängen kann.

Für das nächste Labor nehmen wir mit, dass wir uns vorher mehr Zeit für genau ausgearbeiteten Block- und Aktivitätsdiagramme nehmen, damit wir strukturierter und detaillierter im Labor arbeiten können. Doch genau aus diesen ersten Fehlern können wir nun besser vorbereitet in das zweite Labor gehen und haben darüber hinaus bereits einige Erfahrungen mit Microchip Studio und SimulIDE gesammelt, wodurch wir den ATmega 328P besser kennenlernen konnten.

Eigenständigkeitserklärung

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

12.04.2022

Datum

Clara Jädecke

Unterschrift