

LAB 1: I/O PORTS

Informatik und Elektronik

Media Systems

Michael Berens, Prof. Dr. Tessa Taefi

*Anmeldung für das Labor in
myHAW bis zum 15.04.!*

LAB 1: I/O PORTS

Ziel des Labors

Nach diesem Labor sollen sie in der Lage sein

- Analyse:
 - Die verwendete Hardware in einem Blockdiagramm darstellen
- Design:
 - Das für dieses Labor entwickelte Shield in Betrieb nehmen, indem Sie den Schaltplan interpretieren und relevante Auszüge davon dokumentieren
 - Ihre Software in einem UML Aktivitätsdiagramm planen und dokumentieren
- Implementierung:
 - Erste Programme für das Atmel ATmega328P Xplained Mini Board mithilfe des Microchip Studio schreiben und debuggen
 - die I/O Ports des Atmel 328P in Microbit Studio zur Ein- und Ausgabe konfigurieren und nutzen um z.B. einen Pushbuttons einzulesen oder LEDs anzusteuern
- Ihre Ergebnisse fachgerecht zu dokumentieren

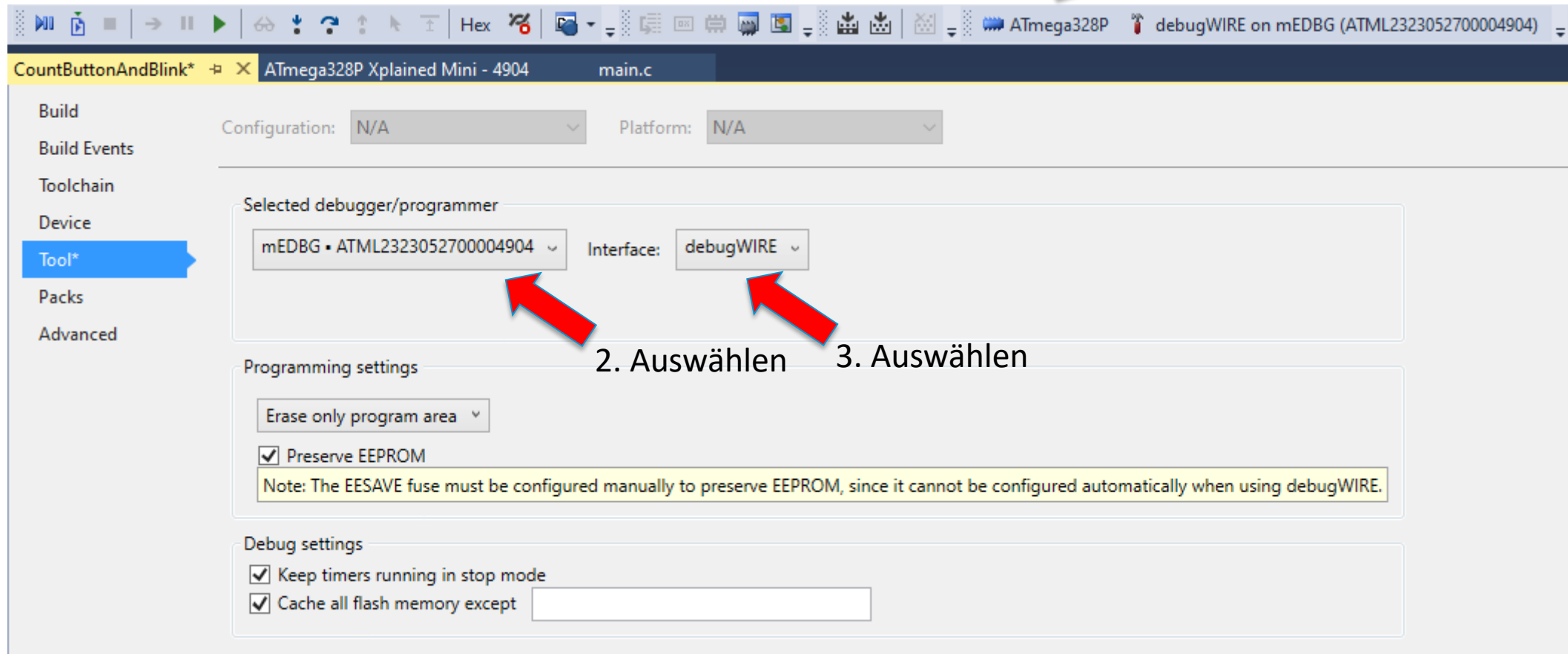
LAB 1: I/O PORTS

Aufgabe 1: Xplained Mini Board

- a. Bringen Sie das Programm **LEDOnButtonPress** aus der Vorlesung 2 auf dem Xplained Mini Board zum Laufen
- b. Designen und schreiben Sie ein neues Proramm namens **CountButtonAndBlink**.
Dieses soll fortlaufend die Anzahl der „Button presses“ zählen und die LED entsprechend oft blinken lassen.
 - D.h. nach dem Start des Programm ist die LED aus
 - Wird der Button gedrückt, blinkt die LED ein Mal
 - Wird der Button noch einmal gedrückt, blinkt die LED zwei Mal
 - Wird der noch einmal gedrückt, blinkt die LED drei Mal, ...
 - Sie können davon ausgehen, dass der Button nicht gedrückt wird, während die LED noch am Blinken ist, d.h. ein Button press während der Blinkdauer soll nicht gezählt werden

HINWEISE: NEUE SOLUTION ANLEGEN

Embedded Debugger



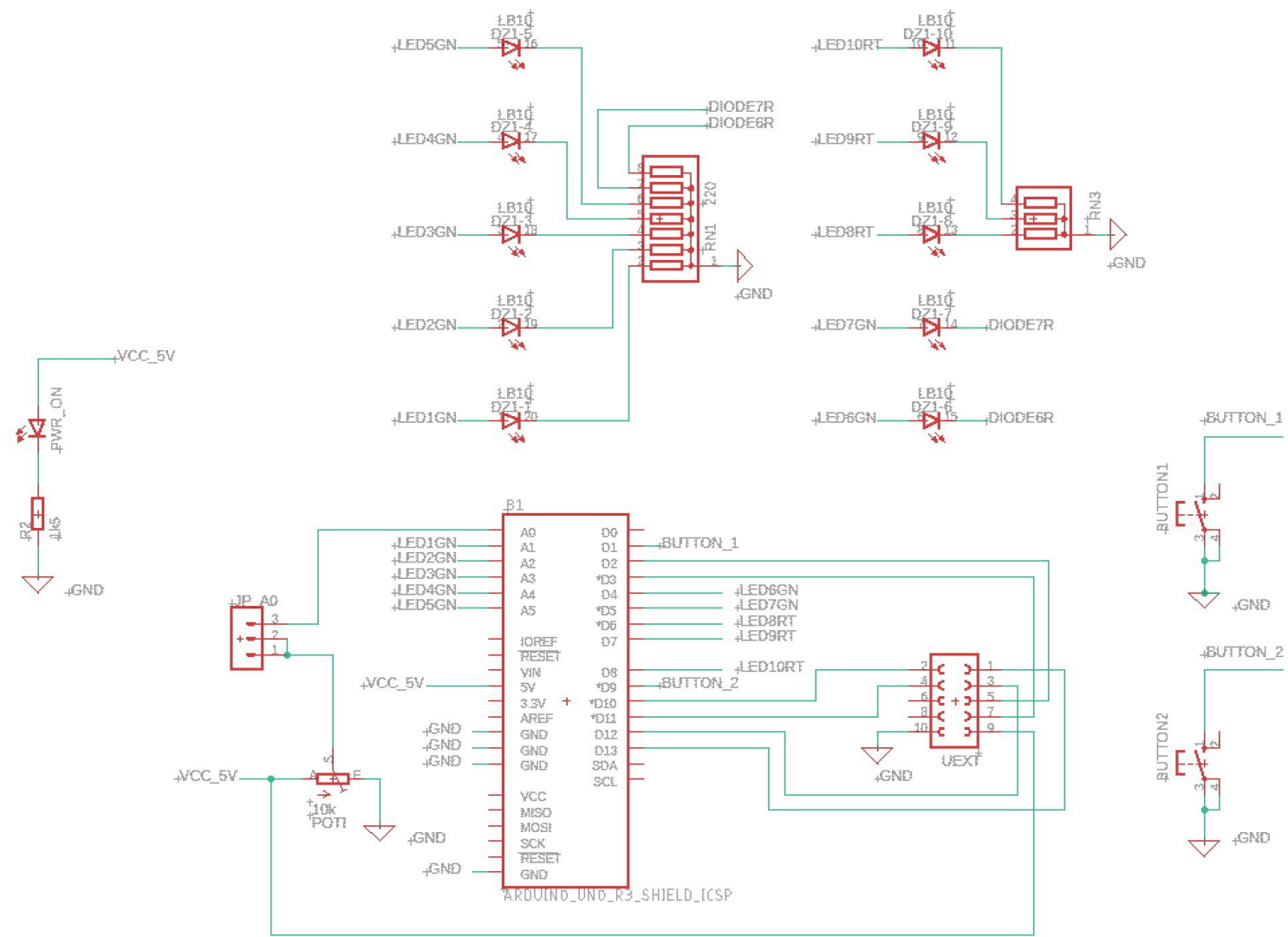
LAB 1: I/O PORTS

Aufgabe 2: Shield in Betrieb nehmen

Entfernen Sie die USB Verbindung, so dass das Xplained Board stromlos ist und stecken Sie vorsichtig das selbstentwickelte Shield auf.

- a. Betrachten Sie den Schaltplan auf der folgenden Slide, und berechnen Sie, wie viel Strom eine LED benötigt, sowie die gesamte LED Bank benötigt. Ist der Strom den die Ports zur Verfügung stellen können ausreichend?
- b. Portieren Sie Ihr Programm **LEDOnButtonPress** auf die neue Situation mit dem Shield, so dass es mit Button1 gesteuert werden kann, und die erste grüne LED blinkt
- c. Experimentieren Sie damit, was passiert wenn Sie den Pullup Widerstand des Push-Buttons entfernen. Notieren sie Ihre Beobachtungen im Bericht.
- d. Erweitern Sie das Programm, so dass mithilfe eines Drucks auf Button 2 die Anzahl der Blinks zurück auf 0 gesetzt werden kann
- e. Ändern Sie ihr Programm so, dass Statt zu Blinken, die Anzahl der gezählten Button presses als vorzeichenlose Dualzahl ausgegeben wird (LSB = LED1GN). Die maximal darstellbare Zahl soll 15 sein. Fangen Sie bei einem Überlauf wieder von 0 an zu zählen und erlauben Sie den Reset mit Button 2.

SELBSTENTWICKELTES SHIELD



Bauteil	Atmega328P
LED1GN	PC1
LED2GN	PC2
LED3GN	PC3
LED4GN	PC4
LED5GN	PC5
LED6GN	PD4
LED7GN	PD5
LED8RT	PD6
LED9RT	PD7
LED10RT	PD0
Button1	PD1
Button2	PB1
Potentiometer	PC0

LAB 1: I/O PORTS

Aufgabe 3: Interrupts

Entwickeln Sie ein Programm, dass

- alle grünen LED anschaltet, solange Button 1 gedrückt wird
 - Alle roten LED anschaltet, solange Button 2 gedrückt wird
 - Setzen Sie geschickt Macros ein, um ihnen die Programmierarbeit zu erleichtern und den Code lesbarer zu machen
- a. Verwenden Sie Polling
 - b. Verwenden Sie
 - für Button 1 einen Pin Change Interrupt
 - Für Button 2 einen externen Interrupt