## Aufgabenblatt 03 Termine: KW 18, KW 19

| Gruppe | |
|---|---|
| Name(n) | Matrikelnummer(n) |
| | |

## 3 Entprellen von Eingangssignalen

Bei der Bearbeitung der Aufgaben 1.5 und 1.6 des ersten Aufgabenblattes sollte Ihnen aufgefallen sein, dass eine einzige Betätigung des Tasters in vielen Fällen zu mehreren Zustandswechseln der LED geführt hat. Und auch in diesen Fällen sind Ihnen wiederum nur die Tasterbetätigungen mit einer ungeraden Anzahl an Zustandswechseln als Fehlfunktion aufgefallen.
Dieses Verhalten ist auf das **Prellen** (de.wikipedia.org/wiki/Prellen) des Tasters zurückzuführen. Dabei kommt es im Inneren des Tasters zu mehrfachem Kontaktschluss, bedingt durch mechanische Eigenschaften, Abnutzungserscheinungen, etc.

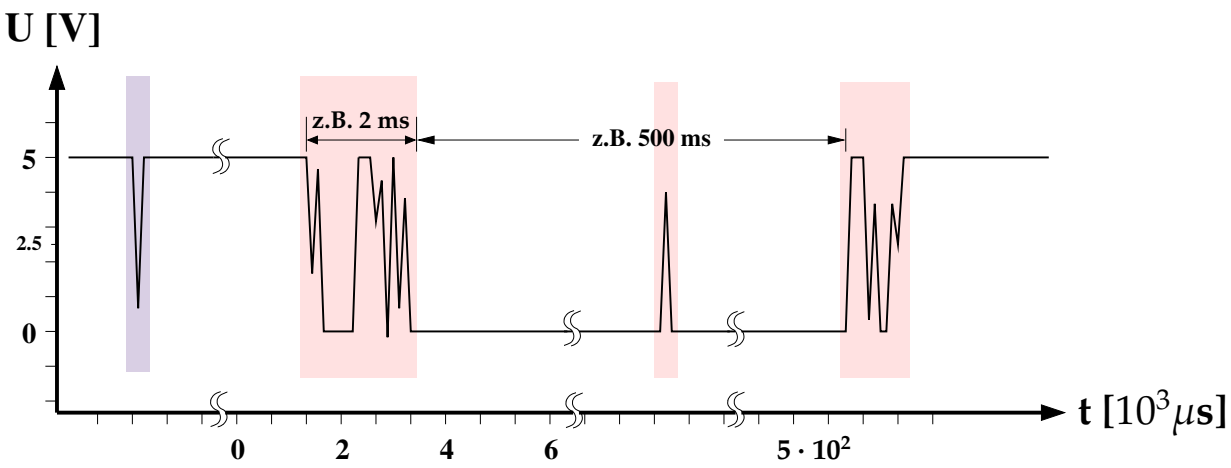


Abbildung 1: Signalverlauf / Prellen eines Tasters.

Abbildung 1 zeigt exemplarisch die Ausgabespannung eines realen Tasters, wenn dieser, wie in Aufgabe 1 eingeführt, mit einem Pullup-Widerstand gegen $V_{CC}$ verschaltet ist. Die pink hinterlegten Bereiche werden durch das nicht ideale Verhalten des Tasters bzw. des Kontaktmaterials im Übergangsbereich und auch z. T. im bereits kontaktierten Zustand hervorgerufen. Der distelfarben hinterlegte Spike ist hingegen eher auf eine unsaubere Versorgungsspannung zurück-

zuführen. Aber letztendlich darf keine dieser kurzzeitigen Änderungen des Spannungspegels als Betätigung des Tasters fehlinterpretiert werden.
Ist die zeitliche Auflösung des digitalen Systems fein genug, so wird das Prellen des Tasters – mehrere Spikes im Signalverlauf (siehe auch Abbildung 2) - als mehrfaches Betätigen interpretiert. Da Taster bei eingebetteten Systemen ein beliebtes Eingabegerät darstellen, sollen Sie im ersten Teil dieses Aufgabenblattes eine Lösung für das Problem entwickeln.

Das Entprellen eines mechanischen Tasters kann sowohl in Hardware als auch in Software realisiert werden. Üblicherweise werden beide Verfahren kombiniert. Ihre Aufgabe ist es, eine Software-basierte Lösung zu entwickeln.

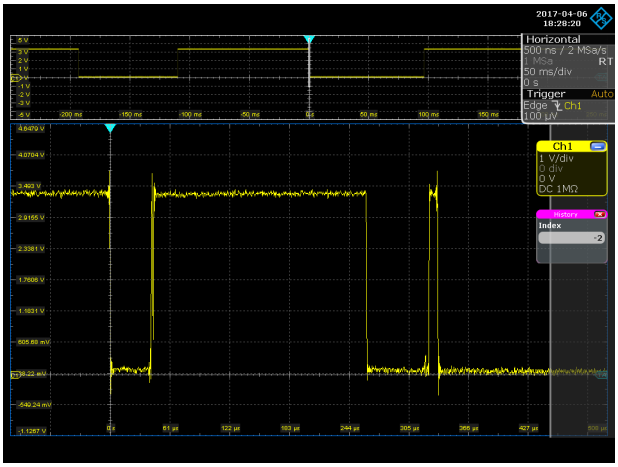


Abbildung 2: Oszillogramm des Signalverlauf am realen Taster. Der Taster prellt hier über ca. 350 $\mu$s, was bereits recht kurz ist. Je nach Bauart und Größe des Tasters/Schalters ist mit Prellzeiten bis in den zweistelligen Millisekunden Bereich zu rechnen.
Sollten Sie beispielsweise in Ihrer Lösung zu Augabe 1.6 lediglich die fallende Flanke zur Erkennung der Tasterbetätigung eingesetzt haben, werden bei obigem Signal mindestens 3 Tasterbetätigungen detektiert, wobei es sich tatsächlich um eine einzelne Betätigung handelt.

**Aufgabe 3.1** Ermitteln der Dauer einer typischen Tasterbetätigung

Ermittlen Sie, wie lange Sie bei der Betätigung eines Tasters diesen typischerweise gedrückt halten.
Hierfür wird es erforderlich werden, das Eingangssignal (quasi)kontinuierlich zu beobachten. Dieses kann durch periodisches, äquidistantes Abtasten des Signals realisiert werden. Dafür lassen sich Hardware-Timer verwenden, welche periodisch einen Interrupt auslösen. Die Aufgabe der dem entsprechenden Interrupt zugeordneten Behandlungsroutine wird dann das Abtasten des Eingangssignals sein.
Beachten Sie hierfür die im folgenden Kapitel: **Hardware-Timer** gegebenen Hinweise.

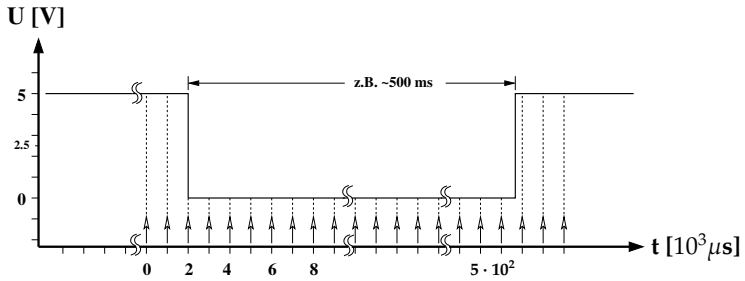


Abbildung 3: Abtasten eines Signals im Abstand 1 mS

## Hardware-Timer

Die folgenden Aufgaben benötigen zur Realisierung Hardware-Timer, die von allen auf den Arduino-Boards eingesetzten Prozessoren in unterschiedlicher Zahl zur Verfügung gestellt werden.
Timer/Counter sind spezielle Hardwareeinheiten der Prozessoren, die, getrieben durch die Systemclock, deren Frequenz ggf. heruntergeteilt werden kann, ein dem jeweiligen Timer fest zugeordnetes Zähl-Register inkrementieren. Die Timer können über Zählfrequenz, Startwert, Vergleichswert und verschiedene Interruptbedingungen flexibel konfiguriert werden. Die auf dem Atmel ATmega328P basierenden Arduino-Boards wie der Arduino UNO besitzen drei Timer (timer0,1,2), die Arduino MEGA-Boards, die auf den ATmega1280 oder ATmega2560 basieren, verfügen über sechs Timer, und das Arduino DUE-Board mit dem Atmel SAM3X8E ARM-Prozessor über neun Timer. Diese sind sehr flexibel einsetzbar. Die `DueTimer` Bibliothek können Sie von der Webseite der Veranstaltung herunterladen: DueTimer Bibliothek.

Folgender Beispielcode skizziert die Verwendung der `DUE-Timer` Bibliotheken:

```
#include <DueTimer.h>
// Objects for configuration of Arduino Due's hardware timers
DueTimer Timer4;     //instantiate timer object

const uint32_t t4_frequency = 1;                    //set timer frequency [Hz]

void timerISR(void) {
 //code of ISR
}
```

```
void setup() {
  // configuration of hardware timer
  if (Timer4.configure(t4_frequency, timerISR)) {
    // Syntax: configure(timer_frequency , callback_function)
    // DUE-Timer is configured, but has not been started
    ; }
  else  { // timer configuration fails...
    ; } //if possible prepare an error message

  Timer4.start();    //normally done on demand by application
}  //end setup

void loop() {
  //code of main loop
}   //end loop
```

Um das skizzierte Beispiel compilieren zu können, müssen Sie die DueTimer Bibliothek in den dafür von der Entwicklungsumgebung vorgegebenen Ordner verschieben. Dieser Ordner befindet sich im Home-Verzeichnis des Benutzers.

| | | |
|---|---|---|
| Der Pfad des Ordners ist unter | Linux: | `~/Arduino/libraries` |
| und unter | Win10/11: | `~\Documents\Arduino\libraries` |

**Aufgabe 3.2** Kriterium zur Erkennung des Tasterzustandes

Für die Entprellung eines digitalen Eingangssignals ist es erforderlich, ein leicht überprüfbares Kriterium für die sichere Erkennung des LOW-Pegels und des HIGH-Pegels vorliegen zu haben. Definieren Sie für die Entprellung eines Tasters ein Kriterium für die sichere Erkennung des betätigten und auch des entlasteten Tasters.
Hierfür wird die die kontinuierliche bzw. die quasi kontinuierliche Beobachtung des Eingangssignals erforderlich werden. Die Überwachung Ihrer Kriterien wird die Erkennung eines vollständigen Betätigungszyklus (gedrückt + losgelassen) des Tasters erfordern.
Bei diesem Lösungsansatz wird es erforderlich werden, das Eingangssignal in für die Entprellung relevanten Zeitbereichen (quasi)kontinuierlich zu beobachten. Dieses kann durch periodisches, äquidistantes Abtasten des Eingangssignals realisiert werden, wie es bereits in Aufg. 3.1 für die Ermittlung der Betätigungsdauer des Tasters eingesetzt wurde.

Diskutieren und begründen Sie, bei welchem Zustandsübergang des Tasters im Hauptprogramm (der Main-Loop) die bei Tasterbetätigung gewünschte Aktion angestoßen werden sollte. Grundsätzlich ist es hier möglich, die bei Betätigung des Tasters anzustoßende Aktion bereits bei erkannter sicherer Betätigung, oder aber erst nach sicherer Deaktivierung des Tasters anzustoßen. Diskutieren Sie die Vor- und Nachteile beider Varianten.
Sollten Sie sich beispielsweise entschieden haben, die gewünschte Aktion bereits bei Betätigung des Tasters und nicht erst am Ende des Betätigungszyklus auszulösen, so soll nach Erkennung eines konstanten Signalpegels (in diesem Fall: konstanter Logikpegel `LOW` nach einer fallenden Flanke über den entsprechend Ihres Kriteriums definierten Zeitraum) diese Information zur Verarbeitung des Ereignisses an das Hauptprogramm weitergegeben werden, wobei natürlich der Taster bis zum Abschluss der Betätigung überwacht werden muss.

### Aufgabe 3.3 Entprellen eines Tasters

Erweitern Sie den Versuchsaufbau aus dem ersten Aufgabenblatt und erstellen Sie ein Programm, das einen zuverlässigen Zustandswechsel der LED in Abhängigkeit der Tasterbetätigung (Aufgabe 1.5 + 1.6) realisiert. Implementieren einen Zähler erkannter Betätigungen des Tasters.
Wie bereits unter Aufg. 3.2 diskutiert, ist die Überwachung eines vollständigen Betätigungszyklus (gedrückt + losgelassen) des Tasters erforderlich. Hier eignet sich zum Beispiel ein endlicher Automat.

Wählen Sie ausgehend von der Konfiguration des Timers (empfohlen wird eine Frequenz von 1 KHz) einen sinnvollen Zeitraum für die Beobachtung des Signals in den relevanten Zeiträumen. Vergessen Sie nicht, beide Zustandsübergänge des Betätigungsvorgangs für einen vollen Betätigungszyklus zu betrachten (Taster gedrückt + Taster losgelassen).
Unter Berücksichtigung des Nyquist-Shannon-Abtasttheorems müsste das Signal aus Abb. 2 selbst bei Außerachtlassen der Unsauberkeiten im Bereich der Flanken allein wegen des Spikes bei ~330$\mu s$ mit mindestens 100 KHz abgetastet werden. Weshalb erreichen wir selbst ohne vorgeschaltete Entprellung durch zusätzliche Hardware selbst bei einer Abtastrate von 1 kHz eine wirkungsvolle Entprellung?

### Aufgabe 3.4 Gleichzeitige Tasterbetätigung

Um Resourcen zu sparen, werden die Benutzerinterface eingebettete Systeme meist recht einfach gehalten, was sich auch in einer ökonomischen Gestaltung der Eingabemöglichkeiten widerspiegelt. Beispielsweise lässt sich ein dritter Taster einsparen, wenn die „gleichzeitige“ Betätigung zweier bereits vorhandener Taster ausgewertet werden kann.

(a) Erweitern Sie Ihr Programm aus der vorigen Aufgabe um die Entprellung eines zweiten Tasters. Beide Taster sollen, wie bereits diskutiert, einzeln ausgelöst werden können.

(b) Zusätzlich soll auch die Möglichkeit geschaffen werden, eine Doppel- oder Parallelbetätigung beider Taster auszuwerten. Definieren Sie hierfür ein Kriterium, das erlaubt, sowohl eine parallele Betätigung der Taster als auch Einzelbetätigungen zu erkennen bzw. zu unterscheiden.

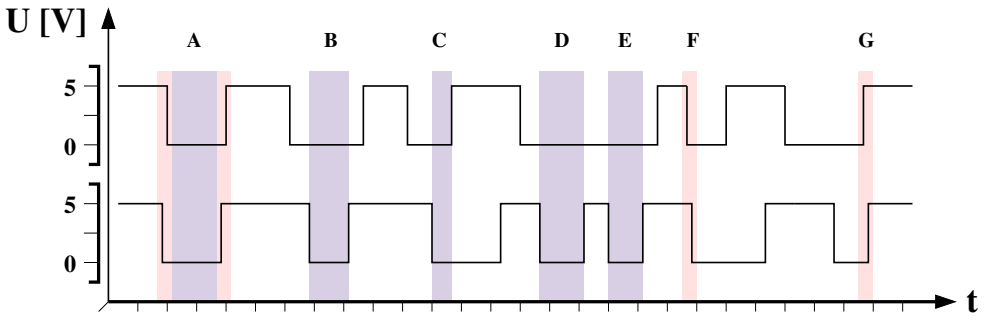


Abbildung 4: Beispielszenarien für Doppel- bzw. Einzelbetätigungen

Hier müssen Sie Fragen diskutieren wie: „Wie definiert sich eine Doppelbetätigung: Beginn in einem festgelegten Zeitfenster, und/oder Ende im festgelegten Zeitfenster, oder ei-

ne einfache Überlappung usw.“. Abbildung 4 zeigt eine kleine Auswahl an Szenarien, die an Hand Ihres Kriteriums eindeutig als Doppel oder Einzelbetätigung klassifiziert werden müssen. Eine Tasterbetätigung darf natürlich auch nur einmal ausgewertet werden: Entweder ist die Betätigung eines Tasters Bestandteil einer Doppelbetätigung (hier stellen die Fälle **D** und **E** ggf. einen Sonderfall dar), oder es ist eine Einzelbetätigung.

(c) Implementieren Sie Ihr Kriterium zur Erkennung einer Doppelbetätigung und zählen Sie die Doppelbetätigungen.