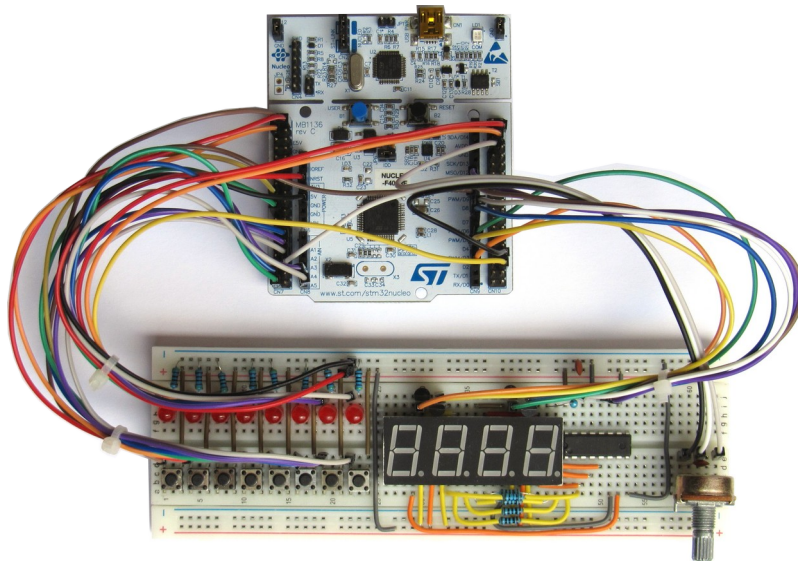


Mikroprozessoren-Labor



Versuch 2, 7-Segment-Anzeige und ADC

Versuchsvorbereitung

Stand: 28. November 2020

I. Versuchsbeschreibung

In diesem Versuch soll die 7-Segment-Anzeige in Betrieb genommen werden.

Danach wird der ADC in Betrieb genommen und der am Eingang gemessene Wert in verschiedenen Varianten angezeigt.

II. Aufgaben

1 Installation und Aufbau

Falls noch nicht geschehen, vervollständigen Sie die Schaltung auf dem Bread-Board.

2 Funktionsweise einer mehrstelligen 7-Segment-Anzeige

2.1 Einstellige Anzeige

Eine einzelne 7-Segment-Anzeige besteht aus 7 „Leucht-Segmenten“ (und einem Dezimalpunkt) die unabhängig voneinander eingeschaltet werden können. Die Segmente bezeichnet man üblicher Weise mit den Buchstaben „a“ bis „g“ in folgender Weise:

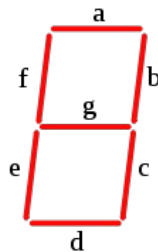


Bild 1: 7-Segment-Anzeige (Quelle: <https://de.wikipedia.org/wiki/Segmentanzeige>)

Abhängig davon, welche Segmente aktiviert werden, kann die Anzeige leicht erkennbar die Ziffern „0“ bis „9“ darstellen.

Wert	x3	x2	x1	x0	Zeichen	a	b	c	d	e	f	g
0	0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	1	0	1	1	0	0	0	0
2	0	0	1	0	2	1	1	0	1	1	0	1
3	0	0	1	1	3	1	1	1	1	0	0	1
4	0	1	0	0	4	0	1	1	0	0	1	1
5	0	1	0	1	5	1	0	1	1	0	1	1
6	0	1	1	0	6	1	0	1	1	1	1	1
7	0	1	1	1	7	1	1	1	0	0	0	0
8	1	0	0	0	8	1	1	1	1	1	1	1
9	1	0	0	1	9	1	1	1	1	0	1	1

Bild 2: Zahlendarstellung (Quelle: <https://de.wikipedia.org/wiki/Segmentanzeige>)

Es gibt auch Ansätze, das komplette Alphabet mit einer 7-Segment-Anzeige darzustellen.

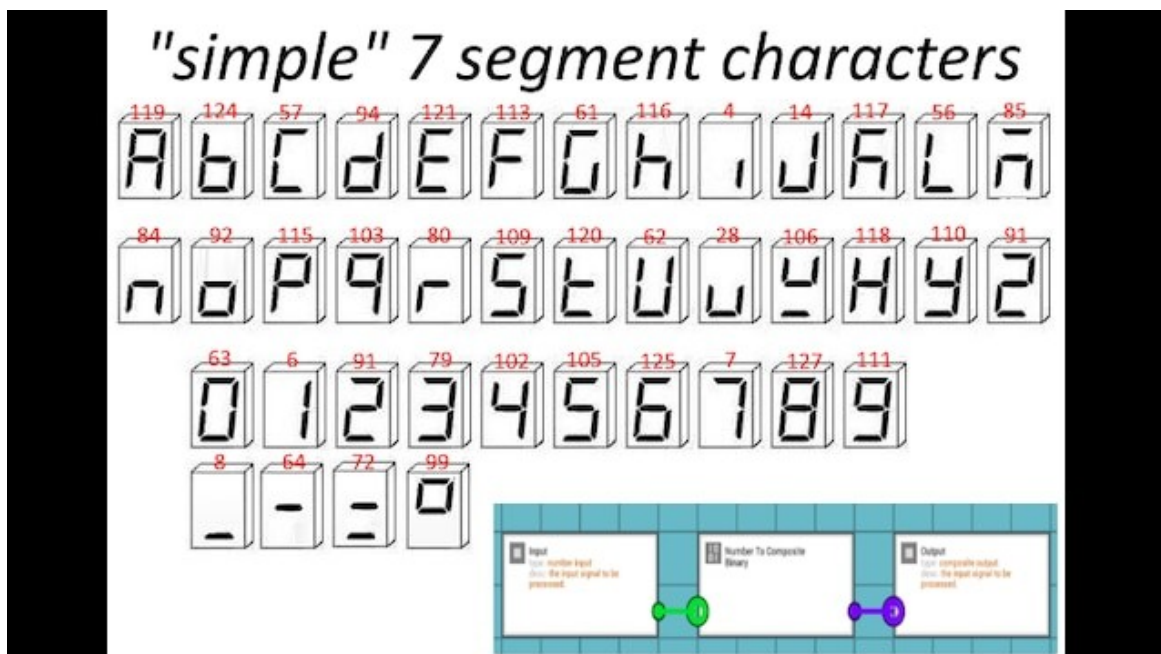


Bild 3: Buchstabendarstellung (Quelle: <https://steamcommunity.com/sharedfiles/filedetails/?id=1701428356>)

Die Darstellung gelingt – ja nach Buchstabe – mehr oder weniger gut. In jedem Fall nutzbar, d. h. eindeutig erkennbar, sind die Buchstaben „A“ bis „F“, so dass eine 7-Segment-Anzeige nicht nur für die Anzeige von Dezimalziffern sondern darüber hinaus für alle Hexadezimalziffern genutzt werden kann.

2.2 Mehrstellige Anzeige

Um die 7-Segmente und den Dezimalpunkt anzusteuern, benötigt man bei einer einstelligen Anzeige 8 Ausgangsleitungen. Das wäre machbar. Für die Ansteuerung einer 4-stelligen 7-Segmentanzeige bräuchte man allerdings bereits 32 Ausgänge und damit mehr als uns z. B. bei dem Nucleo-Board zur Verfügung stehen.

Zeitmultiplexverfahren - Time Multiplexing

Allgemein steuert man daher mehrstellige 7-Segment-Anzeigen im sogenannte **Zeitmultiplexverfahren** an. Dafür werden einerseits die Anoden aller Leuchtdioden einer einzelnen Stelle miteinander verbunden und andererseits die Kathoden der Leuchtdioden der verschiedenen Stellen. Die LEDs befinden sich damit an den Kreuzungspunkten einer 4x8-Matrix, auch wenn das in der nachfolgenden Anzeigestellen-(Digit-)orientierten Darstellung nicht ohne weiteres erkennbar ist.

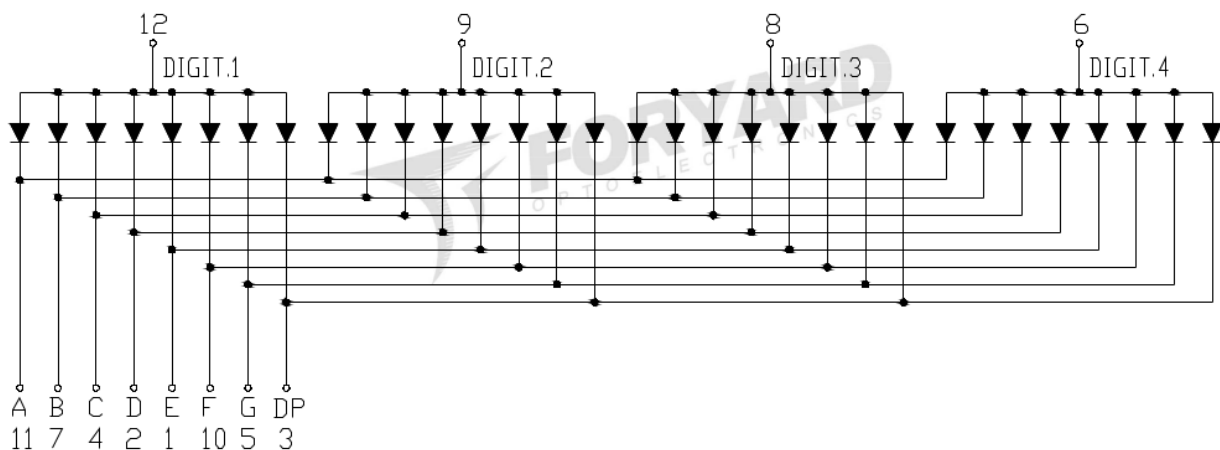


Bild 4: LED-Matrix (Quelle: Datenblatt FYQ-5641BS)

Diese Art der Zusammenschaltung bezeichnet man als „common anode“. Die entsprechenden Verbindungen sind in der mehrstelligen Anzeige des Kits bereits intern hergestellt, so dass nur 12 Anschlüsse nach außen geführt werden müssen.

Selbstverständlich ist alternativ auch der umgekehrte Ansatz bei der Zusammenschaltung möglich. Man kann ebenso gut die Kathoden der Leuchtdioden einer einzelnen Stelle verbinden und die Anoden der verschiedenen Stellen. Diese Schaltung bezeichnet man dann als „common cathode“. Beide Varianten sind kommerziell erhältlich.

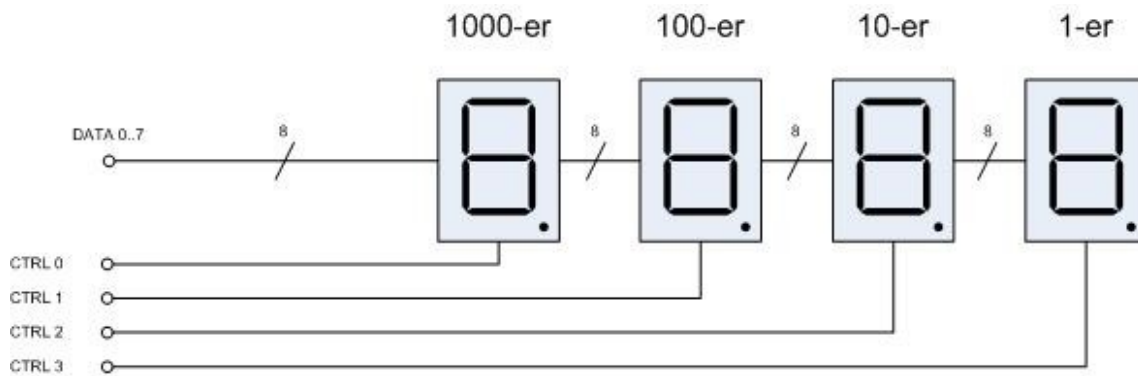


Bild 5: Blockschaltbild 4-stellige 7-Segment-Anzeige für Zeitmultiplexbetrieb

Die Anzeige wird nun so angesteuert, dass immer nur eine der gemeinsamen Anoden (im Blockschaltbild eine der Steuerleitungen CTRL0 bis CTRL3) auf den hohen Spannungspegel gelegt wird. Gleichzeitig werden die Kathoden der LEDs (im Bild DATA 0..7), die bei dieser „aktiven“ Stelle der Anzeige leuchten sollen, über die „Datenleitungen“ auf den niedrigen Spannungspegel gelegt (selbstverständlich unter Verwendung eines Vorwiderstands).

Danach werden die gemeinsamen Anoden der nächsten Stelle (nach CTRL0 wäre das CTRL1) aktiviert (auf hohen Spannungspegel gelegt) und an DATA 0..7 ein niedriger Pegel für die LEDs ausgegeben, die an dieser Stelle leuchten sollen usw.

Aufgrund der Trägheit des menschlichen Auges scheinen alle LEDs dauerhaft zu leuchten, wenn jede Stelle mindestens alle 20 ms (also mit einer Frequenz von mindestens 50 Hz) aktiviert wird.

2.3 Optimierung der Anzahl benötigter μ C-Ausgänge

Mit der im vorangegangenen Abschnitt beschriebenen Technik lässt sich die Anzahl der für eine 4-stellige Anzeige benötigten Ausgänge auf 12 reduzieren. Eine weitere Reduktion benötigter Ausgänge (oder, wenn Sie so wollen, eine Erhöhung der Anzahl der Ausgänge des μ C) kann man durch die Verwendung eines Schieberegisters mit gepufferten Ausgängen erreichen.

Der dem Kit beiliegende Chip SN74HC595 stellt ein solches Schieberegister zur Verfügung.

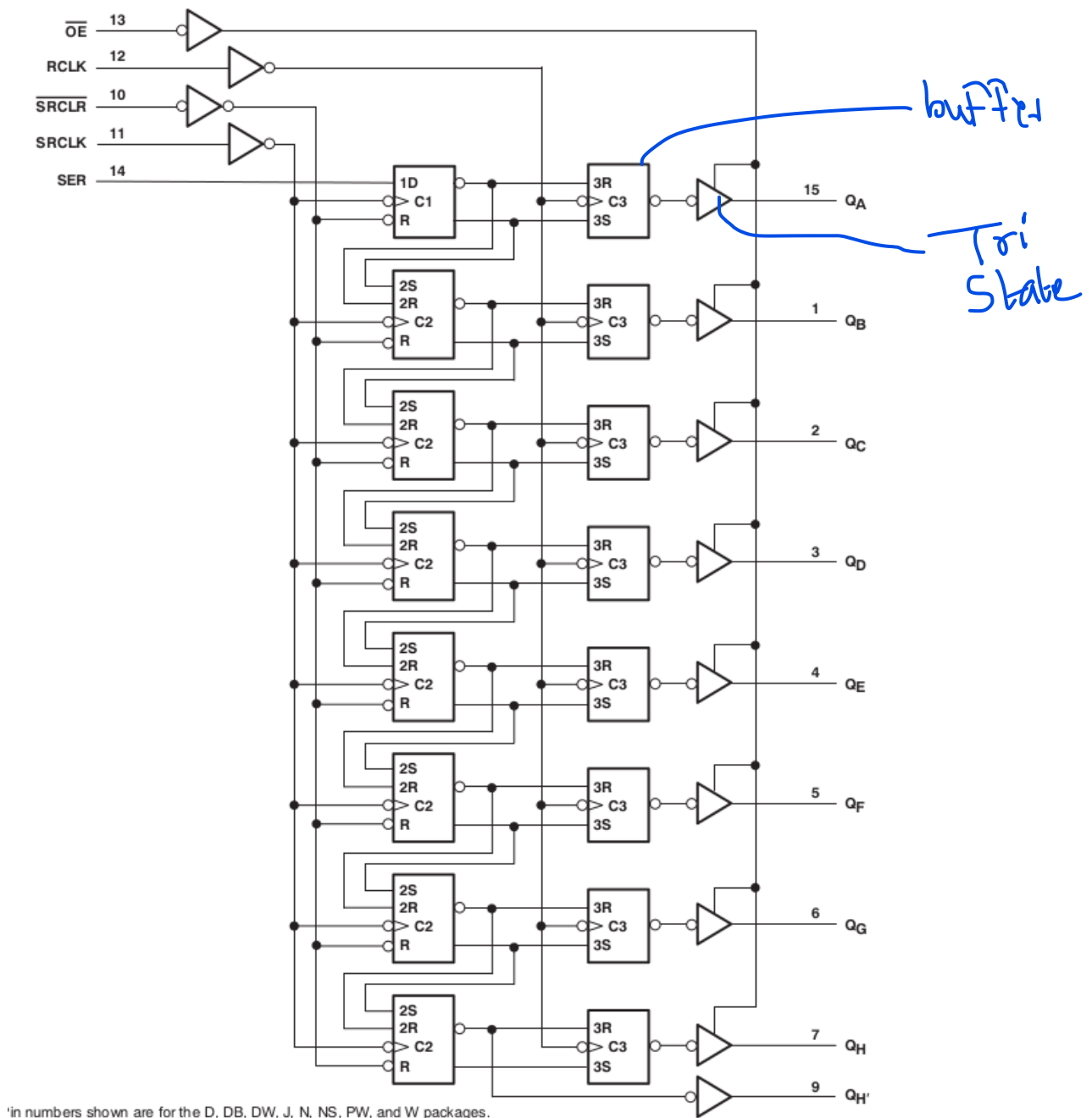


Bild 6: 74HC595 (Quelle: Datenblatt)

Mit jeder steigenden Flanke am Eingang SRCLK werden von jedem im Schaltbild in der linken Spalte dargestellten Flip-Flop (außer dem obersten) die Daten des jeweils darüber dargestellten Flip-Flops übernommen. Das oberste Flip-Flop übernimmt den Wert am Eingang SER. Die Flip-Flops der linken Spalte bilden damit das „eigentliche“ (ungepufferte) Schieberegister. Die Werte an den Ausgängen Q_A bis Q_H ändern sich zunächst nicht, da diese (über einen Tri-State Treiber) mit den Ausgängen der Flip-Flops in der rechten Spalte verbunden sind, die das Pufferregister bilden. Erst bei einer steigenden Flanke am Eingang RCLK werden die Werte der Flip-Flops des Schiebe-

registers in die Flip-Flops des Pufferregisters übernommen und damit an die Ausgänge weitergeleitet.

Um die Werte an den Ausgängen Q_A bis Q_H zu ändern, müssen also die neuen Werte in acht Taktzyklen in das Schieberegister geschoben und dann in das Pufferregister übernommen werden. Damit kann das gepufferte Schieberegister unter Nutzung von drei Ausgängen des μC acht zusätzliche Ausgänge bereitstellen.

In der Schaltung für die Ansteuerung der 4-stelligen 7-Segment-Anzeige werden die Ausgänge des gepufferten Schieberegisters über einen Vorwiderstand mit den Kathoden der Segmente bzw. des Dezimalpunkts verbunden.

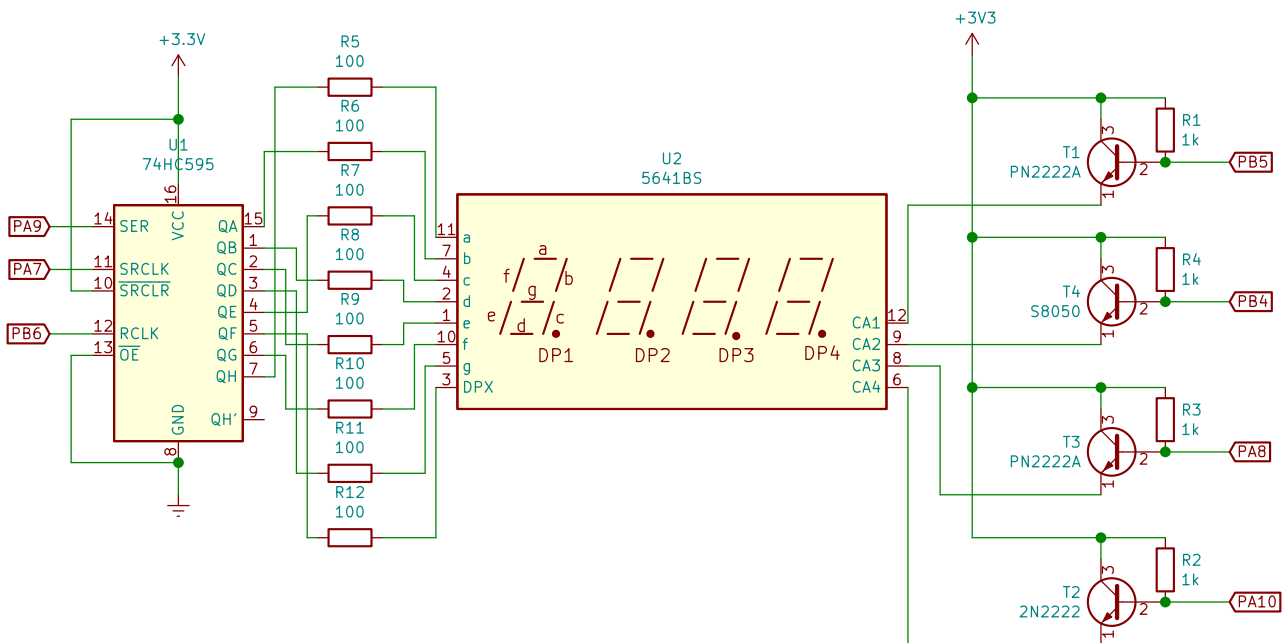


Bild 7: Anschluss der 4-stelligen 7-Segment-Anzeige

Die gemeinsamen Anoden könnten prinzipiell direkt an Ausgänge des μC abgeschlossen werden. Allerdings müsste der Ausgang dann bei der Darstellung einer „8.“ den Strom für 8 „LEDs“ liefern. Mit den gewählten Widerständen läge der Ausgangsstrom zwar noch unterhalb des zulässigen Maximalwertes. Allerdings würde die maximale Ausgangsspannung nicht mehr erreicht, da diese vom Ausgangsstrom abhängt. Damit wäre die Helligkeit von der Anzahl angeschalteter Segmente, d. h. von der dargestellten Ziffer abhängig.



Schriftliche Aufgabe V-1: Ermitteln Sie aus dem Datenblatt des μC die garantierte minimale Ausgangsspannung für einen High-Pegel am Ausgang bei einem Ausgangsstrom von 6 mA und bei einem Ausgangsstrom von 20 mA. Geben Sie die verwendete Tabelle an.

Um den erforderlichen Strom unabhängig von der Anzahl leuchtender Segmente bereitstellen zu können, werden die gemeinsamen Anoden über Transistoren angesteuert. Betreibt man die an die Basen der Transistoren angeschlossenen Ausgänge PB5, PB4, PA8 und PA10 im „Open-Drain“ Modus, sind die Transistoren entweder gesperrt (Basis auf Low-Pegel) oder gehen in die Sättigung

(Basis wird vom Widerstand auf 3,3 V gezogen). In letzterem Fall fällt eine konstante Spannung zwischen der 3,3V Versorgungsspannung und den gemeinsamen Anoden ab. Der Transistor kann unabhängig von der Anzahl eingeschalteter Segmente den benötigten Strom liefern, ohne dass sich der Spannungsabfall erhöht¹.

In Kombination mit der Ansteuerung der Kathoden über das Schieberegister ergibt sich damit bei gerade aktiver Anzeigestelle (Digit) „n“ folgender wiederholter Ablauf bei der Ansteuerung im Zeitmultiplexverfahren:

- Das „Segment-Muster“ für die Anzeigestelle „(n + 1) % 4“ wird in das Schieberegister übertragen.
- Die Anzeigestelle „n“ wird ausgeschaltet.
- „n“ wird auf den Wert „(n + 1) % 4“ gesetzt
- Die Werte für das neue „Segment-Muster“ werden in das Pufferregister übernommen
- Die Anzeigestelle „n“ wird eingeschaltet
- Nach einer Wartezeit ≤ 5 ms (20 ms / Anzahl der Anzeigestellen) wird der Ablauf wiederholt.

✎ **Schriftliche Aufgabe V-2:** Speziell bei unserem Anwendungsfall „Ansteuern der Segmente“ würde auch ein ungepuffertes Schieberegister für die Bereitstellung der zusätzlichen Ausgänge genügen. Überlegen und beschreiben Sie, was bei der Verwendung eines ungepufferten Schieberegisters am oben beschriebenen Ablauf geändert werden muss.

3 Modellierung des Schieberegisters

3.1 Methoden der Klasse implementieren

Im Ausgangsprojekt zum Labor 2 finden Sie die bereits vorbereitete Klasse `CShiftRegister-OutputExtender`. (Denken Sie immer an die Möglichkeit zur Vervollständigung mit Strg+Leertaste in Eclipse.) Implementieren Sie die Methoden entsprechend der Dokumentation in der Header-Datei und den Hinweisen in der cpp-Datei.

3.2 Klasse testen

Schreiben Sie zum Testen der Implementierungen der Klassenmethoden die Funktion `task1()`. Die Funktion weist zunächst den im Ausgangsprojekt vordefinierten `DigitalOut` Objekten `ssdDigit0` bis `ssdDigit3` den Wert 1 oder 0 zu, je nachdem, ob Sie die entsprechende Stelle der Anzeige aktivieren wollen oder nicht (es liegt an Ihnen, welche Stellen Sie aktivieren wollen).

1 Falls Sie schon einmal eine einfache elektronische Schaltung entworfen haben, würden Sie wahrscheinlich PNP-Transistoren nehmen und den Emitter mit 3,3V und den Kollektor mit der gemeinsamen Anode verbinden. Allerdings enthält der Kit nur 3 PNP-Transistoren. Deshalb verwenden wir hier diese etwas ungewöhnlichere Schaltungsvariante.

Erzeugt Sie dann ein Objekt `ssdSegments` vom Typ `CShiftRegisterOutputExtender` unter Verwendung der vordefinierten `DigitalOut` Objekte `ssdSerialData`, `ssdShiftRegisterClock` und `ssdBufferRegisterClock`.

Lesen Sie in einer Endlosschleife die Werte der Taster als `uint8_t` (verwenden Sie den entsprechenden Code aus Labor 1), negieren Sie den Wert bitweise und weisen das Ergebnis dem Objekt `ssdSegments` zu. Das bitweise Negieren ist notwendig, weil die Kathoden der LEDs in der Anzeige mit dem Ausgang des Schieberegisters verbunden sind, d. h. das Segment leuchtet, wenn am Ausgang des Schieberegisters der Low-Pegel anliegt.

Wenn Ihr Testprogramm korrekt läuft, sollte der Druck auf einen Taster dazu führen, dass ein Segment (bzw. der Dezimalpunkt) der Anzeige leuchtet.

✎ **Schriftliche Aufgabe V-3:** Erstellen Sie eine Tabelle, die dokumentiert, welches Bit (0 bis 7) des Wertes, den Sie `ssdSegments` zuweisen, welchem Segment der Anzeige („a“ bis „g“ bzw. Dezimalpunkt) zugeordnet ist.

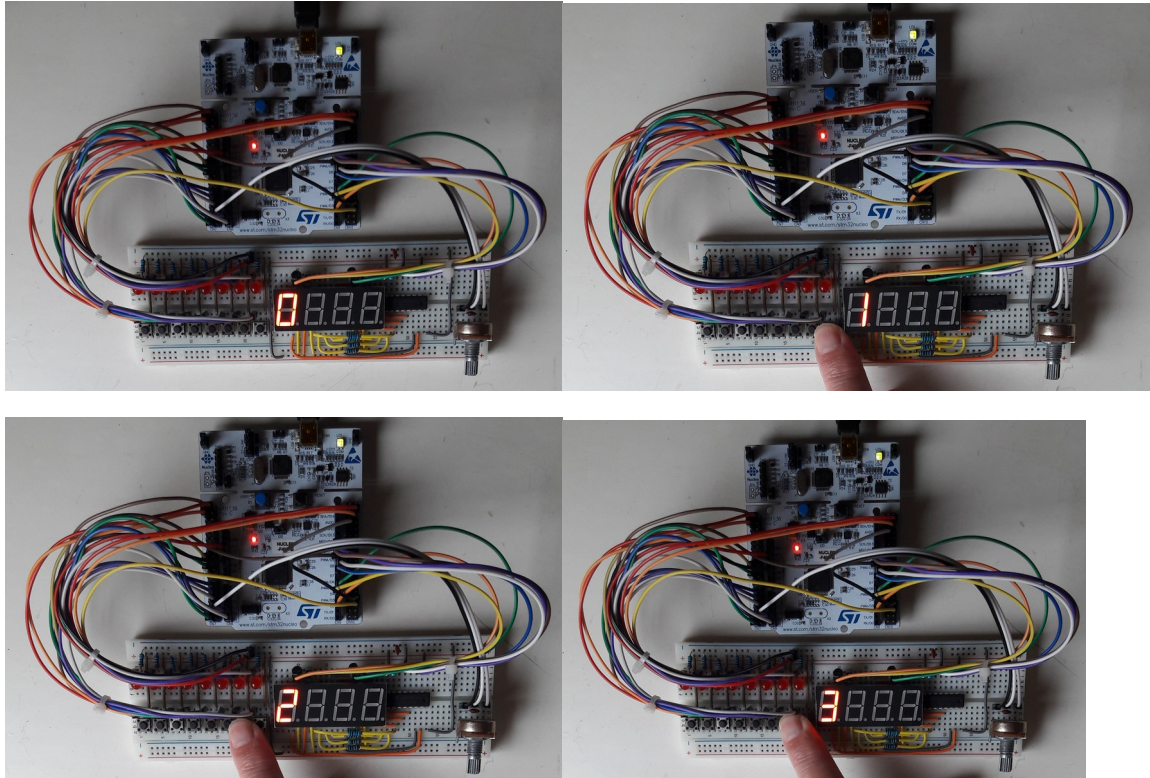
3.3 Erstellen der Segmenttabelle

Erstellen Sie die Funktion `task2()`. Kopieren Sie die Implementierung von `task1` und nehmen Sie folgende Änderungen vor:

1. Erstellen Sie ein Array `uint8_t patterns[16]` als lokale Variable. Initialisieren Sie das Array so, dass es die Index-Werte 0-9 auf die entsprechenden 7-Segment-Muster für die Ziffern 0-9 und die Werte 10-15 auf die Segment-Muster für die Buchstaben A-F abbildet.
2. Ersetzen Sie die Zuweisung des von den Tastern eingelesenen Bit-Musters an `ssdSegments` durch folgende Anweisungen²:
 - a) Lesen Sie das Bit-Muster von den Tastern. (Um sicherzustellen, dass Sie das Bit-Muster korrekt einlesen, sollten Sie es auch auf den LEDs über den Tastern ausgegeben, vergl. Labor 1)
 - b) Wählen Sie von dem Wert nur die vier niederwertigsten Bit aus (bitweise Und-Verknüpfung mit der entsprechenden Maske).
 - c) Verwenden Sie den resultierenden Wert als Index für einen Zugriff auf das Array `patterns` und weisen Sie den ausgelesenen Wert dem Objekt `ssdSegments` zu.
3. Testen Sie Ihre Tabelle.

² Die Anweisungen sind komplizierter als die Lösung. Das Ganze kann als eine Zuweisung mit einem nicht sehr komplizierten Ausdruck auf der rechten Seite implementiert werden.

- ✎ **Schriftliche Aufgabe V-4:** Dokumentieren Sie Ihren erfolgreichen Test durch 4 Aufnahmen, **auf denen auch Ihr Studentenausweis (lesbar!) zu sehen ist**. Die Aufnahmen müssen die Tests für die letzten vier Ziffern Ihrer Matrikelnummer dokumentieren.



3.4 Geschwindigkeit der zusätzlichen Ausgänge

Damit die 7-Segment-Anzeige „flackerfrei“ betrieben werden kann, muss – wie oben bereits ausgeführt – sichergestellt sein, dass mindestens alle 5 ms die aktive Anzeigestelle gewechselt wird. Das kann nur funktionieren, wenn die Übertragung neuer Daten an die Ausgänge des Schieberegisters ausreichend schnell erfolgt. Die für die Übertragung erforderliche Zeit können Sie mit dem Logik-Analysator messen.

Verbinden Sie (gleiches Vorgehen wie bei Versuch 1) drei Eingänge des Logik-Analysators mit den Ausgängen des μC , die die Schieberegistereingänge SER, SRCLOCK und RCLOCK ansteuern. Welche das sind können Sie dem Schaltplan oben entnehmen und wo die entsprechenden Pins auf dem NUCLEO-Board zu finden sind der Dokumentation des Boards, die Sie auch schon beim Aufbau der Hardware verwendet haben.

Starten Sie PulseView und stellen Sie „2 k Samples“ und „24 MHz“ ein. Nutzen Sie, wie im folgenden Ausschnitt dargestellt, die Möglichkeit, die Bezeichner an den Eingangssignalen umzubenennen, damit klar ist, welche Kurve in der Darstellung welches Signal repräsentiert.

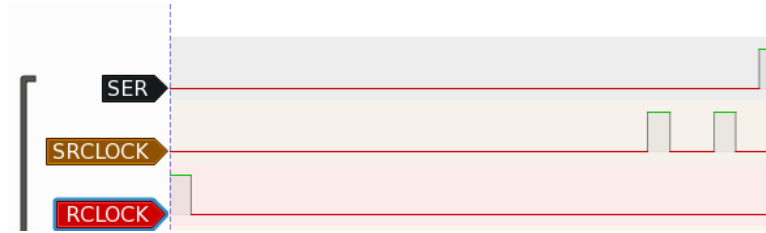


Bild 8: Ausschnitt PulseView-Messung

- ✎ **Schriftliche Aufgabe V-5:** Nehmen Sie eine Messung für die Darstellung der Ziffer „0“ vor (keine Taste gedrückt). Kopieren Sie einen Screen-Shot des vollständigen PulseView-Fensters in Ihre Labordokumentation. Bestimmen Sie aus Ihrer Messung die Zeit, die benötigt wird, um neue Werte an die Ausgänge des Schieberegisters zu übertragen. Wie können Sie sinnvoll abschätzen, welche Zeit für das Einlesen der Taster benötigt wird? Wie könnten Sie die Zeit genauer bestimmen?