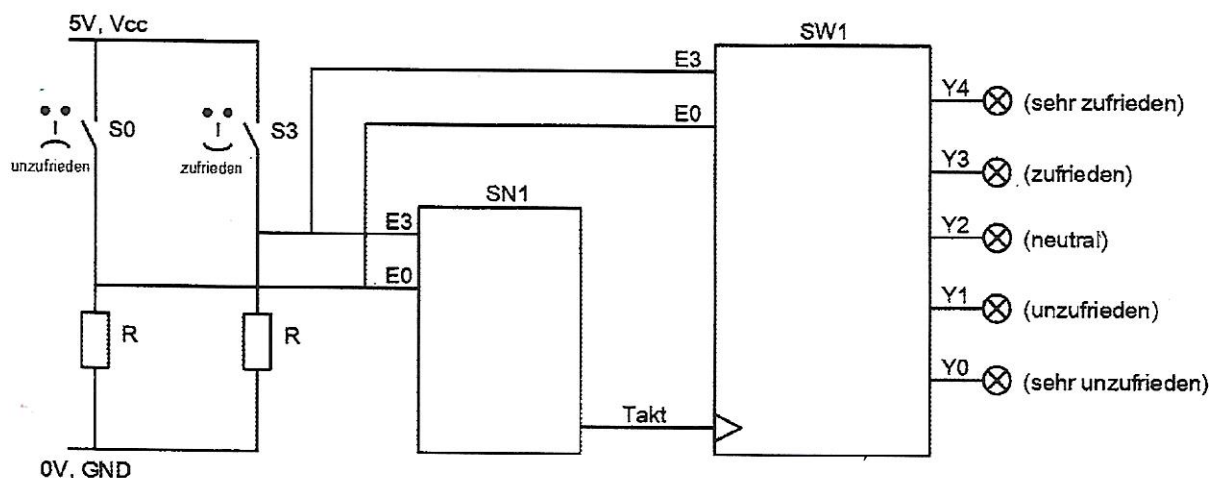


## Aufgabe 2 Automat zur Erfassung der Kundenzufriedenheit in einem Lebensmittelmarkt

In einem Lebensmittelmarkt wird am Ausgang die Zufriedenheit der Kunden mit Hilfe eines Automaten ermittelt. Hierzu existieren Schaltflächen mit unterschiedlichen Symbolen, sogenannte „Smileys“. Die Kunden drücken hierbei auf dasjenige Symbol, welches am ehesten Ihrem Einkaufserlebnis entspricht. Diese Benutzereingaben werden für die Belegschaft und die Geschäftsleitung zur Qualitätskontrolle entsprechend ausgewertet.



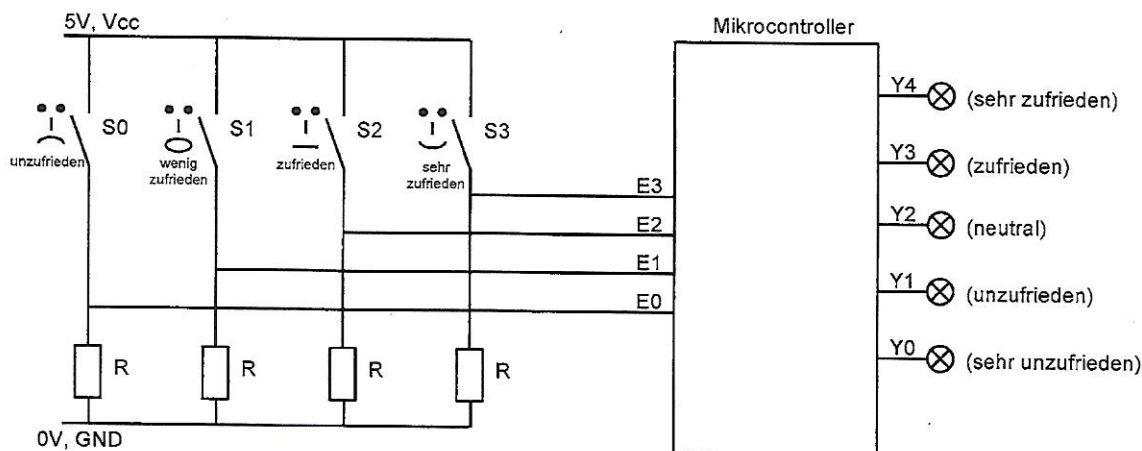
Direkt nach dem Einschalten leuchtet zunächst die Lampe Y2 für "neutral". Wird der Taster S3 betätigt steigt die Zufriedenheit um eine Position nach oben. Maximal bis "sehr zufrieden" erreicht ist. Wird der Taster S0 betätigt sinkt die Zufriedenheit um eine Position nach unten bis irgendwann das Minimum "sehr unzufrieden" erreicht ist.

Hinweis: Die Taster S0 und S3 werden niemals gleichzeitig betätigt.

### 2.1 Schaltwerk mit Codierer

- |       |   |   |
|-------|---|---|
| 2.1.1 | Sobald einer der beiden Taster S0 bzw. S3 betätigt wird, liefert das Schaltnetz SN1 eine positive Takt-Flanke. Ermitteln Sie die Funktionsgleichung für das Takt-Signal.  | 1 |
| 2.1.2 | Zeichnen Sie das Zustandsdiagramm für das Schaltwerk SW1.   | 3 |
| 2.1.3 | Geben Sie an, wieviel Speicherelemente das Schaltwerk SW1 mindestens haben muss. Begründen Sie ihre Entscheidung.   | 1 |
| 2.1.4 | Das Schaltwerk SW1 soll mit minimaler Speicherzahl realisiert werden. Erstellen Sie die codierte Zustandsübergangstabelle.  | 4 |
| 2.1.5 | Ein Umcodierer im Schaltwerk SW1 erzeugt die Signale mit denen die Zufriedenheits-Lampen angesteuert werden. Dieser nutzt als Eingangssignale die Ausgangssignale der Zustandsspeicher. Erstellen Sie die Wahrheitstabelle für den Umcodierer | 2 |

## 2.2 Lösung mit Mikrocontroller



Damit die Zufriedenheit der Kunden differenzierter ausgewertet werden kann, haben diese nun die Auswahl von 4 verschiedenen Rückmeldungen.

Diese Auswertung erfolgt über die globale 8-Bit Variable "zufriedenheit", welche ein Maß für die erzielte Zufriedenheit darstellt.

Auswirkung der Tasten S3..S0.

Wird S3 betätigt, so wird die "zufriedenheit" um 2 erhöht, allerdings nicht über 255 hinaus.

Wird S2 betätigt, so wird die "zufriedenheit" um 1 erhöht, allerdings nicht über 255 hinaus.

Wird S1 betätigt, so wird die "zufriedenheit" um 1 vermindert, allerdings nicht unter 0.

Wird S0 betätigt, so wird die "zufriedenheit" um 2 vermindert, allerdings nicht unter 0.

Der ermittelte Grad der "zufriedenheit" wird durch die Lampen Y4 .. Y0 dargestellt. Dieser gehorcht folgender Gesetzmäßigkeit

$225 < \text{zufriedenheit} \leq 255$	→ sehr zufrieden
$156 < \text{zufriedenheit} \leq 225$	→ zufrieden
$100 < \text{zufriedenheit} \leq 155$	→ neutral
$30 < \text{zufriedenheit} \leq 100$	→ unzufrieden
$0 \leq \text{zufriedenheit} \leq 30$	→ sehr unzufrieden

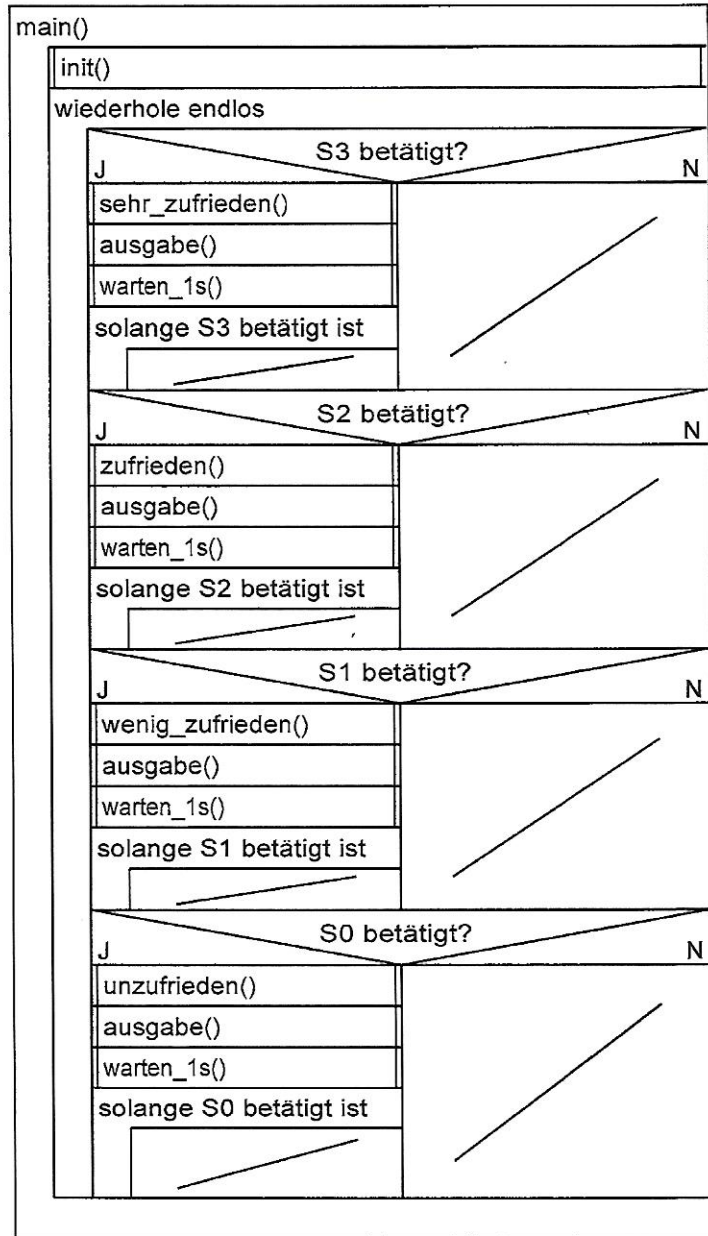
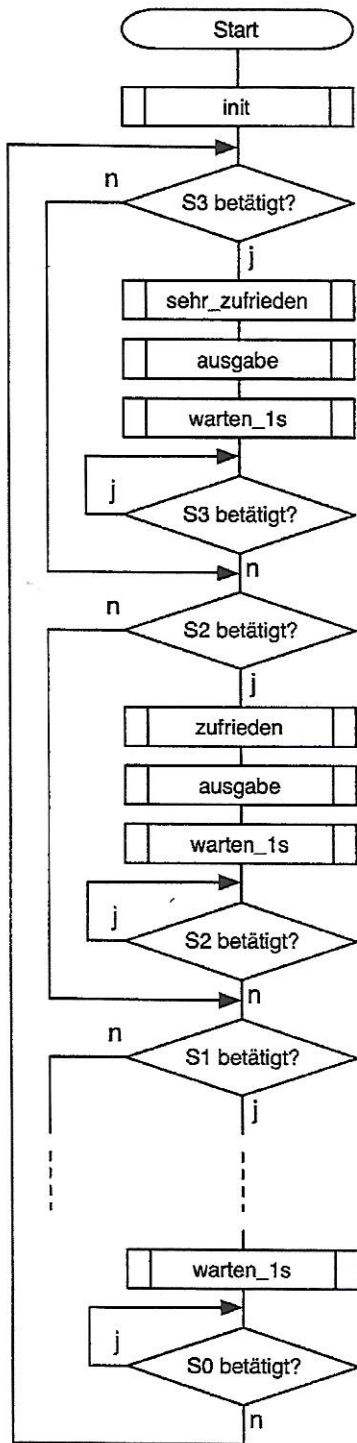
### Hinweise zum Struktogrammen bzw. PAP für das Hauptprogramm

Das Unterprogramm **ausgabe** steuert die Ausgänge Y4..Y0 in Abhängigkeit von der Variable "zufriedenheit". Hat diese z.B. den Wert 200, so soll Y3 leuchten, die anderen Ausgänge müssen Low-Pegel annehmen.

Das Unterprogramm **sehr\_zufrieden** wird bei Betätigung des Tasters S3 aufgerufen und verändert die "zufriedenheit" nach obigen Regeln. Analoges gilt für die Unterprogramme **zufrieden**, **wenig\_zufrieden** und **unzufrieden**.

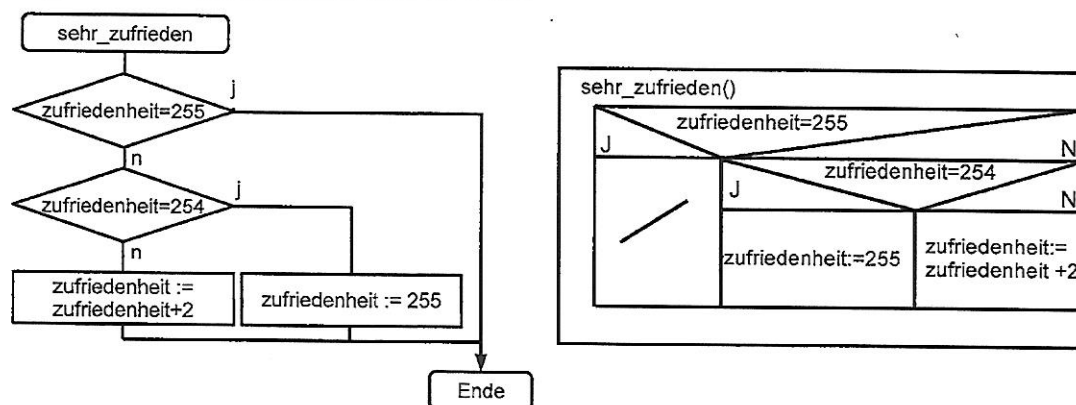
Das Unterprogramm **warten\_1s** erzeugt eine zeitlich Verzögerung von 1s Dauer. Hierzu wird ein 16-Bit Timer im Interruptbetrieb genutzt.

Der Mikrocontroller benötigt für einen Maschinenzyklus eine Mikrosekunde.





- 2.2.1 Ordnen Sie den Signalen E3 .. E0 und Y4 .. Y0 passende Portpins des MC für das an ihrer Schule eingesetzte Mikrocontrollersystem zu. 2
- 2.2.2 Erstellen Sie das Unterprogramm **init** in C bzw. Assembler. Hier wird der Anfangswert für die "zufriedenheit" mit dem Wert 128 gestartet, auf den Ausgängen Y4...Y0 wird „neutral“ ausgegeben. Zusätzlich wird der Timer konfiguriert und die dazu gehörige Interrupt Quelle freigegeben. 2
- 2.2.3 Beschreiben Sie detailliert die Abfrage der Taster im Hauptprogramm (Siehe Programmablauf bzw. Struktogramm). Gehen Sie insbesondere auf das Verhalten des Programms ein, wenn eine Taste dauerhaft betätigt wird oder die Tasten in sehr kurzen Abständen betätigt werden, also jemand auf den Tasten „trommelt“ bzw. wenn die Tasten prellen. 3
- 2.2.4 Setzen Sie den Programmablaufplan bzw. das Struktogramm für das Unterprogramm **sehr\_zufrieden** in Assembler oder in C um. 4



- 2.2.5 Erstellen Sie das Struktogramm oder den Programmablaufplan für das Unterprogramm **ausgabe**. 3
- 2.2.6 Beschreiben Sie, wie Sie mit Hilfe des 16-Bit-Timers im Interruptbetrieb und einer zusätzlichen 8-Bit-Variable "anzahl" die gewünschte Zeitverzögerung von 1s Dauer im Unterprogramm **warte\_1s** realisieren können. Ermitteln Sie die Einstellwerte für den Timer, sowie Start- und Endwert für die Variable "anzahl". 3
- 2.2.7 Setzen Sie die Interrupt-Service-Routine für den 16-Bit Timer der in 2.2.6 beschriebenen Lösung in C oder Assembler um. 2