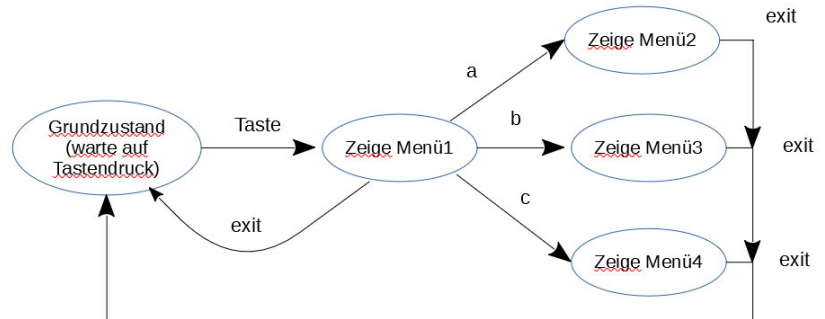


Testat-Aufgabe: Komfortable Uhr/Stoppuhr

Aufgabe:

Es soll eine Uhr gebaut werden, die auch eine Stoppuhr Funktion hat und über **Menüs** eingestellt werden kann. Die Menü-Punkte werden mittels Drehgeber angewählt und durch Knopfdruck ausgewählt. Es soll 2 Menü-Ebenen geben, wie im Zustandsdiagramm rechts gezeigt ist.

Sie bekommen hierzu die Klasse ScrollMenu (im gleichnamigen Modul von Runtime) gestellt, die die Anzeige, Auswahl und Löschen von Menüs erledigt. Sie müssen die Menüs lediglich konfigurieren und anwenden.



Menü1:

- Setze Uhrzeit (ruft Menü2 auf (a))
- Setze Stoppuhrzeit (ruft Menü 3 auf (b))
- Einstellungen (ruft Menü4 auf (c))
- Exit (kehrt zum Grundzustand zurück und löscht Menüanzeige)

Menü2 (Uhr) und Menü3 (Stoppuhr):

- Setze Stunde bei Uhr / Stoppuhr (ruft Einstellfunktion für Stundenwert auf und löscht Menü)
- Setze Minute bei Uhr / Stoppuhr (ruft Einstellfunktion für Minutenwert auf und löscht Menü)
- Setze Sekunde bei Uhr / Stoppuhr (ruft Einstellfunktion für Sekundenwert auf und löscht Menü)
- Exit (kehrt zum Grundzustand zurück und löscht Menüanzeige)

Menü4:

- Wähle 12h Anzeige-Format (nur Uhrzeit – 12h Anzeige mit Angabe „am“ oder „pm“)
- Wähle 24h Anzeige-Format (nur Uhrzeit)
- Exit (kehrt zum Grundzustand zurück und löscht Menüanzeige)

Menüs sind Objekte der Klasse **ScrollMenu** und müssen konfiguriert werden – im Wesentlichen heißt das, die Texte für die Menüeinträge anzulegen. Hier ein Beispiel wie das für Menü1 aussehen kann:

```
// Menue 1
static const char entry11[] PROGMEM = " Uhr einstellen      ";
static const char entry12[] PROGMEM = " Stoppuhr einstellen";
static const char entry13[] PROGMEM = " Anzeigeart aendern ";
static const char entry14[] PROGMEM = " Menue beenden      ";
static const char clear1[]  PROGMEM = "                    ";
static const char title1[]  PROGMEM = "--FUNKTION WAEHLEN--";
static const char* const menu1[] PROGMEM =
    {clear1, title1, entry11, entry12, entry13, entry14};
```

Beachten Sie, dass alle Texte im Flash liegen müssen – alles andere wäre auch extrem unwirtschaftlich! Die Texte, welche zu einem Menü gehören, werden in einem Zeiger-Array zusammengefasst, welches ebenfalls im Flash liegt. Dabei ist die Reihenfolge bedeutsam: der erste Eintrag (hier clear1) verweist auf einen String mit dem eine Menüzeile gelöscht werden kann. Der zweite Eintrag verweist auf die Überschrift des Menüs (hier title1). Ab dem dritten Eintrag kommen die eigentlichen Menüpunkte (entry11 bis entry14). Die Menütexte sollten alle gleiche Länge haben, damit das Überschreiben bzw. Löschen ohne „Rückstände“ auf dem Display funktioniert.

Der Konstruktor des ScrollMenu Objektes bekommt dieses Array als ersten Parameter übergeben. Damit kann das Objekt dann die Menüanzeige und Auswahl generieren. Damit das Objekt die Größe des Menüs kennt, müssen wir ihm die Anzahl der Einträge als zweiten Parameter mitteilen, beispielsweise so: `(sizeof(menu1)/sizeof(char*))`.

Ein Menü-Objekt soll die Anzeige und Auswahl komplett abwickeln. Daher muss es das Display und den Drehgeber kennen. Wir übergeben daher Zeiger auf je ein LCD-Objekt und ein QuadEncoder-Objekt. Diese beiden müssen also vor dem Menü-Objekt bereits erzeugt worden sein. Ferner müssen wir noch angeben, wo auf dem Display die Menü-Anzeige erfolgen soll. In der Grafik unten wurde die rechte Hälfte des 40x4 Displays dafür vorgesehen (Start ab Spalte 20). Hier nun ein Beispiel, wie die 3 Objekte (disp, quenc und men1) erzeugt und verbunden werden können:

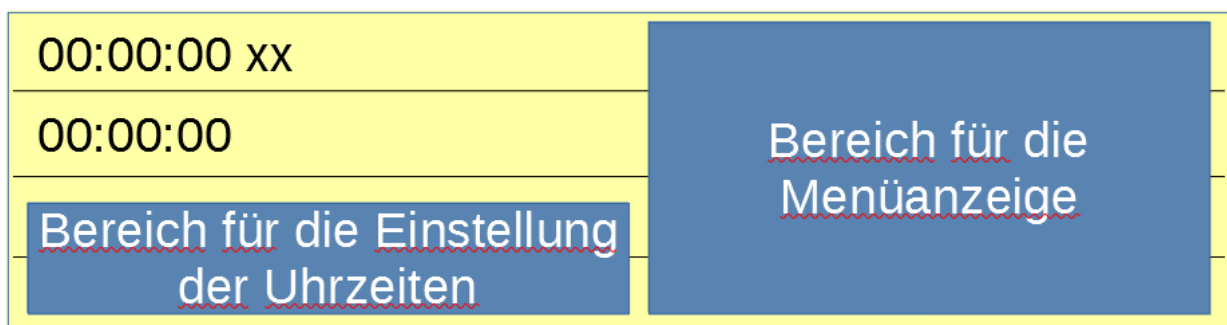
```
static LCD disp(PC, LCD_Type_40x4, DISPLAY_ON|CURSOR_OFF|WRAPPING_OFF|BLINK_OFF);
static QuadEncoder quenc(PJ);
// Erzeuge ein Menue mit den in den menue-Array menu1 verlinkten Eintraegen. Zeige es
// vierzeilig beginnend ab Spalte 20 auf dem Display disp an.
// Nutze quenc für die Menueauswahl
static ScrollMenu men1(menu1, (sizeof(menu1)/sizeof(char*)), &disp, &quenc, 4, 20);
```

Nach dem Muster von `men1` können auch die ScrollMenu-Objekte Menüs2 bis 4 erzeugt werden.

Die Verwendung eines Menüs ist nun denkbar einfach: die Methode `select()` erledigt alles. Sie ist eine Funktion, die die Nummer des schließlich gewählten Menüpunkts zurückliefert. Der erste Menüpunkt (im Beispiel `entry11`) wird dabei als null-ter Eintrag gezählt:

```
switch (men1.select()) {
    case 0: // hier Menü2 aufrufen
    case 1: // hier Menü3 aufrufen
    case 2: // hier Menü4 aufrufen
    case 3: // exit - nothing to do;
}
```

Die Verwendung des Displays (40x4) ist im folgenden Bild gezeigt. Die Uhrzeit-Anzeige soll im Grundzustand sekundlich aktualisiert werden. Die Stoppuhr, sofern sie läuft, ebenfalls. Die Position „XX“ zeigt die Texte „am“ oder „pm“ sofern 12h Anzeige Format gewählt wurde. Die Menütexte müssen natürlich in den vorgesehenen Bereich hineinpassen: wir haben ab Spalte 20 noch 20 Zeichen zur Verfügung. Wenn Sie längere Texte definieren wollen, muss eine Startspalte weiter links liegend gewählt werden.



Definieren Sie eine Taste, die die Stoppuhr startet/stoppt. Eine andere Taste soll das Menü 1 aufrufen. Sobald Stunden, Minuten, oder Sekunden-Einstellung (aus Menü2 oder Menü3 heraus) ausgewählt wurde, können Sie bspw. in den unteren beiden Zeilen des Displays die Zahleneingabe mittels QuadEncoder programmieren. Alternativ können Sie aber auch direkt die Anzeigen in Zeile 0 und Zeile 1 verändern (wie das geht, haben Sie in Übung 6 schon gelernt!). Natürlich wäre eine entsprechende Aufforderung auf dem Display an den Nutzer sinnvoll, damit er weiß, was von ihm erwartet wird.

Nutzen Sie ggf. auch die LEDs, um Tastenechos anzuzeigen.

Überlegen Sie, ob nicht eine Klasse Uhr geeignet wäre sowohl Uhr als auch Stoppuhr einmalig zu implementieren und dann 2 Objekte davon zu erzeugen.

Bereitgestellte Software:

Verwenden Sie die Bibliothek Runtime und davon die folgenden Module. Sehen Sie sich die DoxyGen Dokumentation dazu an!

Basics – einfach nur inkludieren und die darin enthaltenen Definitionen (z.B. Handles für Ports und Timer nutzen).

DigiPort – abstrahiert die Nutzung von Ports als digitale E/A. Die Tasten sind an Port K angeschlossen. LEDs an Port A. Es ist Ihnen überlassen, ob Sie für die Tasten mit einem DigiPortRAW Objekt arbeiten wollen und selbst polling am Port durchführen, oder ob Sie DigiPortIRPT und eine ISR zur Reaktion auf Tasten verwenden wollen.

QuadEncoder – Die Klasse QuadEncoder ist eine Spezialisierung der Klasse DigiPortIRPT und kann die speziellen Signale des Drehgebers (sogenannte quadratur-encodierte Signale) über Interrupts erkennen und interpretieren. Dabei wird dem Encoder der Wertebereich mitgegeben aus dem er beim Drehen des Knopfes Werte generieren soll. Es gibt eine set Methode mit der man diesen Bereich jederzeit ändern kann (für Stundeneingabe: 0..23, für Sekunden und Minuteneingabe: 0..59). Der Drehgeber ist an Port J angeschlossen.

LCD – Verwenden Sie die einfache LCD Klasse (nicht LCDS), um selbst die Positionierung auf dem Display zu kontrollieren. Das LCD ist an Port C angeschlossen.

Im Modul **ScrollMenu** ist die gleichnamige Klasse enthalten.

Timer – verwenden Sie einen Timer Ihrer Wahl für die Implementierung der Zeitbasis der Uhr.

Es ist Ihnen überlassen, ob Sie für die Lösung der Aufgabe Multi-Tasking verwenden möchten. Dann benötigen Sie auch das Modul **OSKernel**. Sofern Sie OSKernel nicht verwenden, müssen Sie die Interrupts selbst aktivieren: sei(), sonst dürfen Sie es nicht!

Die diversen Einstellmodi (Stunden, Minute, Sekunden) mit dem Drehgeber sind ähnlich – denken Sie über eine universelle Einstellfunktion nach. M.a.W.: halten Sie den Code kurz und vermeiden Sie „copy-paste“ sondern setzen Sie parametrisierte Unterprogramme ein! Die Lösung zu dieser Aufgabe hat keine komplexe Struktur – denken Sie zu Beginn gut über Ihre Programmstruktur nach! Erst dann Programmieren.

Für die Erlangung des Testats müssen Sie:

1. **Eine fehler- und warnungsfrei übersetzbare, funktional korrekte, vollständige und lauffähige Version Ihres Programms als Quellcode abgeben.**
2. **Ein Ablaufdiagramm erstellen und abgeben.**
3. **Die Funktionsweise des Programms erklären können.**

Die Lösung der Aufgabenstellung erfolgt individuell – keine Gruppenabgabe.

Zur Teilnahme am Testat müssen Sie **in QIS angemeldet sein! Ohne gültige QIS-Anmeldung zum Zeitpunkt der Abgabe wird das Testat nicht gewertet!**

Beachten Sie die An- und Abmeldezeiträume – Ausnahmen davon gibt es nicht!

Noch ein paar Tipps / Fragen zum Testen:

- Was passiert mit der sekundlichen Aktualisierung der Zeit-Anzeige während Menüs angezeigt werden? Läuft die Aktualisierung weiter oder soll sie bis zum Ende der Menüeingabe stehen bleiben? (Idealerweise sollte sie weiterlaufen).
- Werden die Zeitanzeigen immer korrekt aktualisiert? Wenn bspw. die Stoppuhr nicht läuft und der Nutzer die Stoppuhr Zeit einstellt, wird dann die Anzeige trotzdem korrekt aktualisiert?
- Wenn die Umschaltung 12/24h Anzeige erfolgt: wird dann die Uhr korrekt aktualisiert.
- Was passiert, wenn der Nutzer die Menüeingabe nicht beendet?