

Thema: Speicher

Aufgabe 1

Verändern Sie Ihre Implementierung aus Übungszettel 3, Aufgabe 2 (oder wahlweise Aufgabe 1) folgendermaßen:

- a) Entfernen Sie die DIP-Schalter-Anschlüsse zur Einstellung des Startwertes
- b) Integrieren Sie die Treiberschicht für die Serielle Schnittstelle aus Übungszettel 4
- c) Übermitteln Sie über die serielle Schnittstelle ein benutzungsfreundliches Menü, mit dem der Benutzer (über das Terminalprogramm) am PC die folgenden Dinge einstellen kann:
 - a. Stoppuhr starten
 - b. Stoppuhr stoppen, es soll dann auf dem Terminal die gestoppte Zeit ausgegeben werden
 - c. Startzeit einstellen
 - d. Aktuell eingestellte Startzeit ausgeben

Die Zeichenketten, die Sie für das Menü über die serielle Schnittstelle ausgeben müssen, sollen als Konstanten im Flash angelegt werden.

Abgabe Aufgabe 1: Vorführung

Aufgabe 2

Verändern Sie die Implementierung aus Aufgabe 1 so, dass Sie die Startzeit, die über das Terminal im Menü eingegeben wurde, dauerhaft im EEPROM speichern. Vergeben Sie in der Programmierung für die Variable im EEPROM einen sinnvollen Initialwert.

Da dieser Initialwert auch auf den Controller geschrieben werden muss, müssen Sie das Werkzeug für den Programmdownload verändern. Dies können Sie auf die folgende Art und Weise erreichen:

Sie werden dann nach dem Build-Vorgang im Outputs-Ordner eine neue Datei entdecken: name.eep. Diese Datei enthält den Inhalt des EEPROMs und muss ebenfalls auf den ATmega geflasht werden. Das Werkzeug USBasp kann dies jedoch nicht, daher müssen Sie ein neues sogenanntes „externes Tool“ unter Atmel Studio anlegen. Öffnen Sie dazu das Menü Tools→External Tools. Gehen Sie in das bereits angelegte Tool namens USBasp und kopieren Sie sich dort das „Command“ heraus. Legen Sie anschließend ein neues Tool über „Add“ an, nennen Sie dieses Tool „USBasp + EEPROM“ und fügen sie das eben kopierte Kommando unter „Command“ ein. Danach kopieren sie aus dem USBasp-Tool die „Arguments“ und fügen diese ebenfalls im neuen Tool ein. Anschließend editieren Sie das Feld „Arguments“ und hängen am Ende das folgende an:

```
-U eeprom:w:"$(ProjectDir)Debug\$(TargetName).eep":i
```

Dann noch die Check-Box bei “Use Output window” setzen, Apply drücken und fertig.

Flashen Sie nun nur noch mit diesem neuen Werkzeug, welches zusätzlich zu Ihrem Programm auch die *.eep-Datei mit auf den ATmega flasht. Sie finden das neu erstellte Werkzeug im Tools-Menü.

Abgabe Aufgabe 2: Vorführung

Aufgabe 3

Angenommen, Sie möchten für das EEPROM Wear Leveling implementieren, wie würden Sie dies aufbauen? Wie sähen die Implementierungen von Lese- und Schreibzugriffen aus? Überlegen Sie sich ein Konzept, Implementierung ist nicht gefordert (aber Sie dürfen natürlich gerne ...).

Hinweis 1: wenn Sie eine nächste Speicherstelle auswählen, müssen Sie wissen, welche Speicherstelle sie beim letzten Speichervorgang gewählt hatten. Diese darf natürlich auch nicht an einer einzigen Stelle im EEPROM liegen.

Hinweis 2: Variablen im SRAM können die zuletzt benutzte Speicherstelle nicht aufbewahren, da diese nach Abschaltung der Spannungsversorgung ihren Wert verlieren.

Hinweis 3: es ist hilfreich, über das „Alter“ eines Speicherinhaltes nachzudenken.

Hinweis 4: Speicherstellen, die noch niemals beschrieben worden sind, enthalten den Wert 0xFF.

Abgabe Aufgabe 3: mündlich