

- Nachtrag zur Vorlesung
- Aufgabe 1

1 Lebensdauer von Variablen

- Die Lebensdauer einer Variablen bestimmt, wie lange der Speicherplatz für die Variable aufgehoben wird
- Zwei Arten
 - ◆ Speicherplatz bleibt für die gesamte Programmausführungszeit reserviert
 - statische (`static`) Variablen
 - ◆ Speicherplatz wird bei Betreten eines Blocks reserviert und danach wieder freigegeben
 - dynamische (`automatic`) Variablen

2 Lebensdauer von Variablen (2)

auto-Variablen

- Alle lokalen Variablen sind automatic-Variablen
 - der Speicher wird bei Betreten des Blocks / der Funktion reserviert und bei Verlassen wieder freigegeben
 - ➡ der Wert einer lokalen Variablen ist beim nächsten Betreten des Blocks nicht mehr sicher verfügbar!
- Lokale auto-Variablen können durch beliebige Ausdrücke initialisiert werden
 - die Initialisierung wird bei jedem Eintritt in den Block wiederholt
 - !!! wird eine auto-Variable nicht initialisiert, ist ihr Wert vor der ersten Zuweisung undefiniert (= irgendwas)

3 Lebensdauer von Variablen (2)

static-Variablen

- Der Speicher für alle globalen Variablen ist generell von Programmstart bis Programmende reserviert
- Lokale Variablen erhalten bei Definition mit dem Schlüsselwort `static` eine **Lebensdauer über die gesamte Programmausführung** hinweg
 - ➡ der Inhalt bleibt bei Verlassen des Blocks erhalten und ist bei einem erneuten Eintreten in den Block noch verfügbar
 - !!! Das Schlüsselwort `static` hat bei globalen Variablen eine völlig andere Bedeutung (Einschränkung des Zugriffs auf das Modul)
- Static-Variablen können durch beliebige konstante Ausdrücke initialisiert werden
 - die Initialisierung wird nur einmal beim Programmstart vorgenommen (auch bei lokalen Variablen!)
 - erfolgt keine explizite Initialisierung, wird automatisch mit 0 vorbelegt

U2-2 Aufgabe 1: snake

U2-2 Aufgabe 1: snake

- Schlange bestehend aus benachbarten LEDs
- Länge 1 bis 5 LEDs, regelbar mit Potentiometer (POTI)
- Geschwindigkeit abhängig von Umgebungshelligkeit
- Bewegungsrichtung umschaltbar mit Taster

U2-3 Parameter der Schlange

U2-3 Parameter der Schlange

- Position des Kopfes
 - ◆ Nummer einer LED
 - ◆ Wertebereich [0; 7]
- Länge der Schlange
 - ◆ Ganzzahl im Bereich [1;5]
- Richtung der Schlange
 - ◆ aufwärts oder abwärts
 - ◆ z.B. 0 oder 1
- Geschwindigkeit der Schlange
 - ◆ hier: Durchlaufzahl der Warteschleife

U2-4 Zerlegung in Teilprobleme

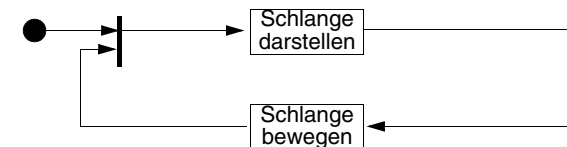
U2-4 Zerlegung in Teilprobleme

- Basisablauf: Welche Schritte wiederholen sich immer wieder?
- Teilprobleme *können* in eigene Funktionen ausgelagert werden
- Wiederkehrende Teilprobleme *sollten* in Funktionen ausgelagert werden
- Welcher Zustand muss über Basisabläufe hinweg erhalten bleiben?
 - ◆ ist Zustand ggf. nur für ein Teilproblem relevant?
 - ◆ Sichtbarkeit dann auf das Teilproblem einschränken (Kapselung)

U2-4 Basisablauf

U2-4 Zerlegung in Teilprobleme

- Darstellung der Schlange
- Bewegung der Schlange



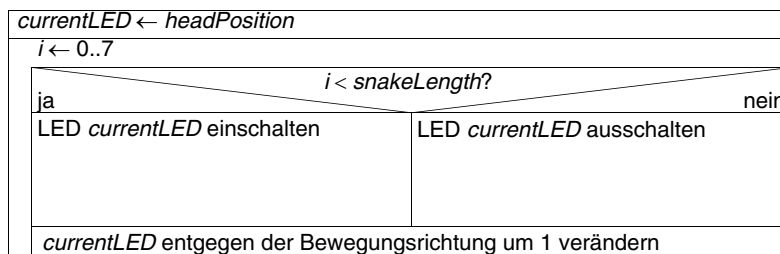
1 Darstellung der Schlange

- Bestimmung der Darstellungsparameter
 - ◆ Kopfposition
 - ◆ Länge
 - ◆ Richtung
- Anzeige der Schlange abhängig von den Parametern
 - ◆ Aktivieren der zur Schlange gehörenden LEDs
 - ◆ Deaktivieren der restlichen LEDs

2 Bewegung der Schlange

- Bestimmung der Bewegungsparameter
 - ◆ Geschwindigkeit
 - ◆ Richtung
- Bewegen der Schlange
 - ◆ Anpassen der Kopfposition abhängig von der Richtung
- Wartepause abhängig von der Geschwindigkeit
- ggf. Richtungsänderung
 - ◆ bisheriger Schlangenschwanz wird zum Schlangenkopf

U2-5 Ablaufplan für Teilproblem Schlangenanzeige



U2-6 Bestimmung der Richtungsänderung

- Der Taster muss periodisch abgefragt werden
 - ◆ man bekommt nur eine Momentaufnahme des Tasters
- evtl. wird ein Tastendruck verpasst
 - ◆ Hochfrequente Abfrage zur Minimierung des Risikos
 - ◆ Wo im Programm ist dies sinnvoll möglich?
 - in der Warteschleife (jedem Durchlauf)
 - hier wird die meiste Zeit verbracht
- mehrfache Abfrage des Tasterzustands während eines Tasterdrucks
 - ◆ Mehrfachinterpretation eines Tasterdrucks vermeiden
 - ◆ Loslassen des Tasters muss explizit registriert werden
 - Historie über den vorigen Tasterzustand mitführen (Flankenerkennung)