**Program variables**:

Variables to store direction state (5)  
Variables to store finger position values (5)  
Variables to store Flexiforce values (5)

**Program constants:**

PWM base frequency   
Extend/retract values for finger directions (e.g. 1 for extend, 0 for retract)  
Values for high/low force thresholds  
Values for finger positions in different modes (pinch, neutral, point…)

**Functions**:

Read Flexiforce value  
Convert Flexiforce value  
Interrupt handlers…

**Calibration**: I’m thinking it’s either a separate state (probably not user-selectable) OR part of the power-up sequence

* Retract fingers
* If “home” interrupt:
  + Save home position – accomplish by resetting timers and counters?
* Extend fingers back to neutral threshold (this step might be optional)
* Stop motors
* Await state change input

**Neutral**: user-selectable state

* Move all fingers to “neutral” position thresholds (predefined constants relative to mechanical zero)
* Stop motors
* (Power saving?)
* Await state change input

**Pinch**: user-selectable state

* Move fingers either to force or predefined position thresholds – cannot remember exactly how pinch is going to work
* Stop motors
* Await state change input

**Handshake**: user-selectable state.

* Optional: wait() (so you don’t have to shove your hand out to time the handshake right)
* Move fingers to force thresholds
* Stop motors & wait
* Extend fingers/release force
* Stop motors
* Await state change input

**Grab**: user-selectable state

* Move fingers to force thresholds
* Maintain force via PID control
* Await state change input

**Point**: user-selectable state

* Move fingers to predefined “point” position thresholds
* Stop motors
* Await state change input

Questions/notes:

* What triggers the change between states/modes? EMG signal or the interface buttons?
* I feel like there are places where the code could implement power-saving measures, but I focused more on the big picture for now.