```
// State Machine for auto-stack
switch (stackStep) {

    // High Stacking
    case HIGH_STACK_START:
        // Only start once we release the button
        if (!controller.get_digital(BTN_ARM_HIGH)) {
            stackStep++;
        }
        break;

    case HIGH_STACK_START + 1:
        // Move the arm up to position
        armSeek = ARM_POS_HIGH;
        // Hold the wrist up
        wristSeek = WRIST_VERTICAL_POS;
        // Wait until arm is above threshold before continuing
        if (armPos > ARM_POS_HIGH - 50) {
            stackStep++;
        }
        break;

    case HIGH_STACK_START + 2:
        // Move wrist to stack cap
        wristSeek = WRIST_BACKWARD_DROP_POS;
        // Wait until wrist in position
        // De-rotate wristpos by armpos to get angle relative to floor
        if (wristPos < WRIST_BACKWARD_DROP_POS + 15 + ( armPos * 3 / 5
         )) {
            stackStep++;
            // Record time this step finished
            timeLastStep = millis();
        }
        break;

    case HIGH_STACK_START + 3:
        // Wait for 0.25 seconds before continuing
        if (timeLastStep + 250 < millis()) {
            stackStep++;
        }
        break;

    case HIGH_STACK_START + 4:
        // Now move arm down
        armSeek = 1;
        // Once it's almost down continue
        if (armPos < ARM_POS_HIGH / 2) {
            stackStep++;
        }
        break;

    case HIGH_STACK_START + 5:
        // Now we are done, clear flags
        armSeek = -1;
        // Hold wrist up though
        wristSeek = WRIST_VERTICAL_POS;
        stackStep = -1;
        break;


// High De-Score
case KNOCK_HIGH_START:
    // Start moving arm up
    armSeek = ARM_POS_HIGH;
    // Move wrist all the way back
    wristSeek = WRIST_BACKWARD_DROP_POS - 100;
    // Wait until arm is up before continuing
    if (armPos > ARM_POS_HIGH) {
        stackStep++;
    }
    break;

case KNOCK_HIGH_START + 1:
    // Now move the wrist forwards to knock cap off
    wristSeek = WRIST_VERTICAL_POS;
    // Wait until wrist is in position
    // Continue last steps of high stack
    if (wristPos > WRIST_VERTICAL_POS - 20) {
        stackStep = HIGH_STACK_START + 3;
        timeLastStep = millis();
    }
    break;


// Low Stacking
case LOW_STACK_START:
    // Wait until button released
    if (!controller.get_digital(BTN_ARM_LOW)) {
        stackStep++;
    }
    break;

case LOW_STACK_START + 1:
    // Move arm up to correct position
    armSeek = ARM_POS_LOW;
    // Hold wrist upright
    wristSeek = WRIST_VERTICAL_POS;
    if (armPos > ARM_POS_LOW - 50) {
        stackStep++;
    }
    break;
```

```
case LOW_STACK_START + 2:
    // Wait until user presses button to continue
    if (controller.get_digital(BTN_ARM_LOW)) {
        stackStep++;
    }
    break;

case LOW_STACK_START + 3:
    // Slowly drop the wrist to place cap
    slowSeek = true;
    wristSeek = WRIST_FORWARD_DROP_POS;
    // Wait until wrist down to continue
    if (wristPos > WRIST_FORWARD_DROP_POS - 15 + ( armPos * 3 /
     5 )) {
        stackStep++;
    }
    break;

case LOW_STACK_START + 4:
    // If wrist button pressed, then go back (user missed pole)
    if (controller.get_digital(BTN_WRIST)) {
        stackStep = LOW_STACK_START + 1;
    }
    // Otherwise, if button pressed again, continue
    if (controller.get_digital(BTN_ARM_LOW)) {
        stackStep++;
    }
    break;

case LOW_STACK_START + 5:
    // Move arm down
    armSeek = 1;
    // Wait until almost down to continue
    if (armPos < ARM_POS_HIGH / 2) {
        stackStep++;
    }
    break;

case LOW_STACK_START + 6:
    // Clear auto-seek flags
    armSeek = -1;
    wristSeek = WRIST_VERTICAL_POS;
    stackStep = -1;
    break;

default:
    // If stackStep has messed up, correct it to -1
    stackStep = -1;
    break;
}
```