Convert the template sketch BOEBot2016.ino to implement a finite state machine driven robot behavior

Disable the GO TO GOAL behavior by commenting out the line (near 208)

```
go_to_goal_idx(go_to_state);
```

Also disable the check for the end of the GO TO GOAL behavior. Comment out (near 236):

```
// If goal is reached advance to the next goal, if any left
  if (go_to_done) {
    if (go_to_point.last == 0) {
      go_to_state++;
      go_to_goal_idx(go_to_state);
    }
}
```

Disable Ultrasonic Sensor back and forth scanning if you will control it in your FSM (near 285)

```
old_angle = scanner_angle;
 scanner_angle += scanner_increment;
if (scanner_angle > 171) {
  scanner_angle = 171 - scanner_increment;
  scanner_increment = - scanner_increment;
} else if (scanner_angle < 9) {
  scanner_angle = 9 - scanner_increment;
  scanner_increment = -scanner_increment;
}
```

Add global state machine variables. If you will be using the ultrasonic distance sensor, add global variables that save the current sensor distance (near line 85)

```
int task_state = 0;
int task_counter = 0;
unsigned long us_last_ping;
unsigned long us_reading1;
unsigned long us_reading2;
int object_count = 0;
```

Initialize your new global variables in setup()

Store the Ultrasonic Sensor distance readings for use by your new state machine (near 252)

```
if (ping_in_progress == 1) {
    us_last_ping = ((unsigned long)60)*((unsigned
long)US_ROUNDTRIP_IN);
… } else {
    us_last_ping = us_ping_result;
```

Write your FSM Code. The best place to put it is in the 100 ms section of the control loop, none of the challenges will require your robot to have a response time less than 100 ms. Every state transition of your FSM thus consumes 100 ms. Line 273.

To control the robot's position use the go_to_goal function, e.g.

```
go_to_goal(0.0, -17.0, 30, 2.0);
```

Then, have your FSM wait for go_to_done to be set, e.g.

```
if (go_to_done == 1) {…
```

To check distances, simply check the variable

```
us_last_ping
```

To rotate the turret, simply set the global variable scanner_angle , e.g.

```
scanner_angle = 45;
```

The servo will be positioned between the current iteration of the 100 ms section and the next