```
1  #include <Servo.h>
2  #include "../vars/constants.h"
3
4  #define SERVO_UPPER_HOMING_POSITION 20
5  #define SERVO_UPPER_HOME_POSITION 0
6  #define SERVO_LOWER_HOME_POSITION 90
7  #define SERVO_HOMING_SPEED 11           //Lower Value -> higher Speed
8  #define SERVO_DEGREES_PER_FUNCTION_CALL 1
9  #define SERVO_MANUAL_TURNING_SPEED 20
10
11 //@brief Object of servo
12 class ServoController
13 {
14     private:
15     Servo servoLower;
16     Servo servoUpper;
17     Servo servoExtender;
18     unsigned long lowerServoLastTurn, upperServoLastTurn;
19
20     public:
21     uint8_t servoLowerPosition = 90;
22     uint8_t servoUpperPosition = 0;
23
24     void init()
25     {
26         pinMode(constants::pins::motor::ServoLowerPin, OUTPUT);
27         pinMode(constants::pins::motor::ServoUpperPin, OUTPUT);
28         pinMode(constants::pins::motor::ServoExtenderPin, OUTPUT);
29         pinMode                                                    ↵
             (constants::pins::motor::ServoExtenderLimitSwitchUpper_Pin,  ↵
             INPUT);
30         pinMode                                                    ↵
             (constants::pins::motor::ServoExtenderLimitSwitchLower_Pin,  ↵
             INPUT);
31         servoLower.attach(constants::pins::motor::ServoLowerPin, 4, 0,  ↵
             180);
32         servoUpper.attach(constants::pins::motor::ServoUpperPin, 5, 0,  ↵
             180);
33         servoExtender.attach(constants::pins::motor::ServoExtenderPin,  ↵
             6, 0, 180);
34
35         resetPosition();
36         homePosition();
37     }
38
39     void applyPosition()
40     {
41         servoUpper.write(servoUpperPosition);
42         servoLower.write(servoLowerPosition);
43     }
44
45     void turnServoUpperCW()
46     {
```

```
47          if(upperServoLastTurn > millis() - SERVO_MANUAL_TURNING_SPEED)
48              return;
49          upperServoLastTurn = millis();
50
51          if(servoUpperPosition > 179)
52          {
53              servoUpperPosition = 180;
54              return;
55          }
56          servoUpperPosition = servoUpperPosition + 1;
57      }
58      void turnServoUpperCCW()
59      {
60          if(upperServoLastTurn > millis() - SERVO_MANUAL_TURNING_SPEED)
61              return;
62          upperServoLastTurn = millis();
63
64          if(servoUpperPosition < 1)
65          {
66              servoUpperPosition = 0;
67              return;
68          }
69          servoUpperPosition = servoUpperPosition - 1;
70      }
71
72      void turnServoLowerCW()
73      {
74          if(lowerServoLastTurn > millis() - SERVO_MANUAL_TURNING_SPEED)
75              return;
76          lowerServoLastTurn = millis();
77
78          if(servoLowerPosition > 179)
79          {
80              servoLowerPosition = 180;
81              return;
82          }
83          servoLowerPosition = servoLowerPosition + 1;
84      }
85      void turnServoLowerCCW()
86      {
87          if(lowerServoLastTurn > millis() - SERVO_MANUAL_TURNING_SPEED)
88              return;
89          lowerServoLastTurn = millis();
90
91          if(servoLowerPosition < 1)
92          {
93              servoLowerPosition = 0;
94              return;
95          }
96          servoLowerPosition = servoLowerPosition - 1;
97      }
98
99      void resetPosition()
```

```
100     {
101         fullyRetractExtender();
102         servoLowerPosition = 90;
103         servoUpperPosition = SERVO_UPPER_HOMING_POSITION;
104         applyPosition();
105     }
106
107     void homePosition()
108     {
109         fullyRetractExtender();
110
111         //Set Upper Servo to Homing Position
112         for (uint8_t i = servoUpperPosition; i !=
              SERVO_UPPER_HOMING_POSITION;)
113         {
114             servoUpperPosition = i;
115             applyPosition();
116             if(i > SERVO_UPPER_HOMING_POSITION)
117                 i--;
118             else
119                 i++;
120
121             delay(SERVO_HOMING_SPEED);
122         }
123
124         //Home Lower Servo
125         for (uint8_t i = servoLowerPosition; i !=
              SERVO_LOWER_HOME_POSITION;)
126         {
127             servoLowerPosition = i;
128             applyPosition();
129             if(i > SERVO_LOWER_HOME_POSITION)
130                 i--;
131             else
132                 i++;
133             delay(SERVO_HOMING_SPEED);
134         }
135
136         //Home Upper Servo
137         for (uint8_t i = servoUpperPosition; i !=
              SERVO_UPPER_HOME_POSITION;)
138         {
139             servoUpperPosition = i;
140             applyPosition();
141             if(i > SERVO_UPPER_HOME_POSITION)
142                 i--;
143             else
144                 i++;
145             delay(SERVO_HOMING_SPEED);
146         }
147     }
148
149     void fullyExtendExtender()
```

```
150     {
151         while(extendExtender());
152     }
153 
154     bool extendExtender()
155     {
156         if(!digitalRead
              (constants::pins::motor::ServoExtenderLimitSwitchUpper_Pin))
157         {
158             Serial.print(millis());
159             Serial.println("  Servo Extender Extend...");
160             servoExtender.write(40);
161             delay(3);
162             servoExtender.write(90);
163         }
164         if(digitalRead
              (constants::pins::motor::ServoExtenderLimitSwitchUpper_Pin))
165         {
166             servoExtender.write(100);
167             while(digitalRead
                  (constants::pins::motor::ServoExtenderLimitSwitchUpper_Pin
                  ))
168             {
169                 ;
170             }
171             delay(3);
172             servoExtender.write(90);
173             return false;
174         }
175         return true;
176     }
177 
178     void fullyRetractExtender()
179     {
180         while(retractExtender());
181     }
182 
183     bool retractExtender()
184     {
185         if(!digitalRead
              (constants::pins::motor::ServoExtenderLimitSwitchLower_Pin))
186         {
187             Serial.print(millis());
188             Serial.println("  Servo Extender Retract...");
189             servoExtender.write(140);
190             delay(3);
191             servoExtender.write(90);
192         }
193         if(digitalRead
              (constants::pins::motor::ServoExtenderLimitSwitchLower_Pin))
194         {
195             servoExtender.write(80);
196             while(digitalRead
```

```
                (constants::pins::motor::ServoExtenderLimitSwitchLower_Pin
                ))
197             {
198                 ;
199             }
200             delay(3);
201             servoExtender.write(90);
202             return false;
203         }
204         return true;
205     }
206 };
```