```cpp
1  #include <Ps3Controller.h>
2  #include "../vars/constants.h"
3
4  class Controller
5  {
6      private:
7      int8_t buttonCirclePrev, buttonXPrev, dPadRightPrev, dPadLeftPrev,  ⮐
         buttonStartPrev;
8      enum KonamiCodeButtons
9      {
10         KONAMI_BUTTON_NONE,
11         KONAMI_BUTTON_UP,
12         KONAMI_BUTTON_DOWN,
13         KONAMI_BUTTON_LEFT,
14         KONAMI_BUTTON_RIGHT,
15         KONAMI_BUTTON_B,
16         KONAMI_BUTTON_A,
17         KONAMI_BUTTON_START
18     };
19     KonamiCodeButtons konamiButtonList[11];
20     public:
21     int8_t joyLX, joyLY, joyRX, joyRY, buttonSelect, buttonStart,      ⮐
         buttonSquare, buttonCircle, buttonTriangle, buttonX, dPadUp,     ⮐
         dPadRight, dPadDown, dPadLeft, joyRButton, joyLButton,           ⮐
         shoulderButtonL, shoulderButtonR, dPadDownPrev, dPadUpPrev;
22     ps3_status_battery battery;
23     uint8_t throttleGas, throttleBreake;
24     unsigned long led_blink_millis;
25     bool led_blink, konamiCode, konamiCodePrev;
26     ps3_status_battery batteryStatusPrev;
27
28
29
30     void init()
31     {
32         Ps3.attachOnConnect(OnConnect);
33         Ps3.begin(constants::controller_vars::BT_ADDR.c_str());
34         for(uint8_t i = 0; i < 11; i++)
35         {
36             konamiButtonList[i] = KONAMI_BUTTON_NONE;
37         }
38         konamiCode = false;
39         konamiCodePrev = false;
40     }
41
42     void loop()
43     {
44         buttonCirclePrev = buttonCircle;
45         buttonXPrev = buttonX;
46         dPadUpPrev = dPadUp;
47         dPadRightPrev = dPadRight;
48         dPadDownPrev = dPadDown;
49         dPadLeftPrev = dPadLeft;
```

```
50          buttonStartPrev = buttonStart;
51          konamiCodePrev = konamiCode;
52
53
54          joyLX = Ps3.data.analog.stick.lx;
55          joyLY = Ps3.data.analog.stick.ly;
56          joyRX = Ps3.data.analog.stick.rx;
57          joyRY = Ps3.data.analog.stick.ry;
58
59          dPadUp    = Ps3.data.button.up;
60          dPadRight = Ps3.data.button.right;
61          dPadDown  = Ps3.data.button.down;
62          dPadLeft  = Ps3.data.button.left;
63
64          joyLButton = Ps3.data.button.l3;
65          joyRButton = Ps3.data.button.r3;
66
67          battery = Ps3.data.status.battery;
68
69          throttleGas    = Ps3.data.analog.button.l2;
70          throttleBreake = Ps3.data.analog.button.r2;
71
72          buttonSelect   = Ps3.data.button.select;
73          buttonStart    = Ps3.data.button.start;
74
75          buttonSquare   = Ps3.data.button.square;
76          buttonCircle   = Ps3.data.button.circle;
77          buttonTriangle = Ps3.data.button.triangle;
78          buttonX        = Ps3.data.button.cross;
79
80          shoulderButtonL = Ps3.data.button.l1;
81          shoulderButtonR = Ps3.data.button.r1;
82
83
84          if(millis() - led_blink_millis > 250)
85          {
86              led_blink_millis = millis();
87              led_blink = !led_blink;
88          }
89
90          if(batteryStatusPrev != Ps3.data.status.battery ||          ⏎
             Ps3.data.status.battery == ps3_status_battery_shutdown)
91          {
92              switch(Ps3.data.status.battery)
93              {
94                  case ps3_status_battery_full:
95                      Serial.println("Controller Battery Full");
96                      Ps3.setPlayer(10);
97                      break;
98                  case ps3_status_battery_high:
99                      Serial.println("Controller Battery High");
100                     Ps3.setPlayer(9);
101                     break;
```

```
102                        case ps3_status_battery_low:
103                            Serial.println("Controller Battery Low");
104                            Ps3.setPlayer(7);
105                            break;
106                        case ps3_status_battery_dying:
107                            Serial.println("Controller Battery Dying");
108                            Ps3.setPlayer(4);
109                            break;
110                        case ps3_status_battery_shutdown:
111                            if(led_blink)
112                            {
113                                Serial.println("Controller Battery Shutdown");
114                                Ps3.setPlayer(4);
115                            }
116                            else
117                            {
118                                Serial.println("Controller Battery Shutdown");
119                                Ps3.setPlayer(1);
120                            }
121                            break;
122                    }
123                }
124            batteryStatusPrev = Ps3.data.status.battery;
125
126            CheckKonami();
127        }
128
129        static void OnConnect()
130        {
131            Ps3.setRumble(100.0, 250);
132            delay(500);
133            Ps3.setRumble(100.0, 250);
134            delay(500);
135            Ps3.setRumble(100.0, 250);
136        }
137
138        void CheckKonami()
139        {
140            if(konamiCode)
141            {
142                konamiButtonList[9] = KONAMI_BUTTON_NONE;
143                konamiCode = false;
144            }
145
146            if(dPadUp && !dPadUpPrev ||
147                dPadDown && !dPadDownPrev ||
148                dPadLeft && !dPadLeftPrev ||
149                dPadRight && !dPadRightPrev ||
150                buttonCircle && !buttonCirclePrev ||
151                buttonX && !buttonXPrev ||
152                buttonStart && !buttonStartPrev)
153            {
154                for(uint8_t i = 0; i < 10; i++)
```

```
155                {
156                    konamiButtonList[i] = konamiButtonList[i+1];
157                }
158            }
159
160        if(dPadUp && !dPadUpPrev)
161            konamiButtonList[10] = KONAMI_BUTTON_UP;
162        else if(dPadDown && !dPadDownPrev)
163            konamiButtonList[10] = KONAMI_BUTTON_DOWN;
164        else if(dPadLeft && !dPadLeftPrev)
165            konamiButtonList[10] = KONAMI_BUTTON_LEFT;
166        else if(dPadRight && !dPadRightPrev)
167            konamiButtonList[10] = KONAMI_BUTTON_RIGHT;
168        else if(buttonCircle && !buttonCirclePrev)
169            konamiButtonList[10] = KONAMI_BUTTON_B;
170        else if(buttonX && !buttonXPrev)
171            konamiButtonList[10] = KONAMI_BUTTON_A;
172        else if(buttonStart && !buttonStartPrev)
173            konamiButtonList[10] = KONAMI_BUTTON_START;
174
175        if(konamiButtonList[10] == KONAMI_BUTTON_START &&
176           konamiButtonList[9] == KONAMI_BUTTON_A &&
177           konamiButtonList[8] == KONAMI_BUTTON_B &&
178           konamiButtonList[7] == KONAMI_BUTTON_RIGHT &&
179           konamiButtonList[6] == KONAMI_BUTTON_LEFT &&
180           konamiButtonList[5] == KONAMI_BUTTON_RIGHT &&
181           konamiButtonList[4] == KONAMI_BUTTON_LEFT &&
182           konamiButtonList[3] == KONAMI_BUTTON_DOWN &&
183           konamiButtonList[2] == KONAMI_BUTTON_DOWN &&
184           konamiButtonList[1] == KONAMI_BUTTON_UP &&
185           konamiButtonList[0] == KONAMI_BUTTON_UP &&
186           konamiCodePrev == false)
187            konamiCode = true;
188    }
189 };
```