

 [leeassam](#) / [arduino-bootcamp](#) Public[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [master](#) ▾

...

[arduino-bootcamp](#) / [QuadSevenSegment_Stopwatch_ShiftRegister](#) /
[QuadSevenSegment_Stopwatch_ShiftRegister.ino](#)

Lee Assam First commit

 0 contributors

218 lines (199 sloc) | 5.4 KB

...

```
1  /*
2  Arduino Bootcamp
3
4  - Operating a Stopwatch with a 4 Digit Segment Display - Part 3
5
6  This project will create a stopwatch with a dual seven segment display.
7  It uses the 74HC595 shift register to control led display with only 3 pins from the Arudino
8
9  created 10/30/2016
10 modified 10/30/2016
11 by: Lee Assam
12 */
13 #include <StopWatch.h>
14 #include <Bounce2.h>
15 //Pins for seven segment LED
16 int segmentPins[] = {2, 3, 4, 5, 6, 7, 8};
17 //LED toggle Pins
18 int displayPins[] = {10, 11, 12, 13};
19 //Global values for display
20 int num1, num2, num3, num4;
21 int displayValue = 0;
22
23 //Start/Stop/Reset Button
24 #define buttonPin 5
25 // Instantiate Bounce object
26 Bounce debouncer = Bounce();
27 unsigned long buttonPressTimeStamp;
28
29 //LED mappings for numbers
30 byte digits[10][8] = {
31     // a b c d e f g .
```

```
32 { 1, 1, 1, 1, 1, 1, 0, 0}, // 0
33 { 0, 1, 1, 0, 0, 0, 0, 0}, // 1
34 { 1, 1, 0, 1, 1, 0, 1, 0}, // 2
35 { 1, 1, 1, 1, 0, 0, 1, 0}, // 3
36 { 0, 1, 1, 0, 0, 1, 1, 0}, // 4
37 { 1, 0, 1, 1, 0, 1, 1, 0}, // 5
38 { 1, 0, 1, 1, 1, 1, 1, 0}, // 6
39 { 1, 1, 1, 0, 0, 0, 0, 0}, // 7
40 { 1, 1, 1, 1, 1, 1, 1, 0}, // 8
41 { 1, 1, 1, 1, 0, 1, 1, 0} // 9
42 };
43
44 //Values for seven segment display
45 byte leds = 0;
46 byte digitVals[10];
47
48 //Shift Register Pins
49 //Pin connected to latch pin (RCLK) of 74HC595
50 #define latchPin 8
51 //Pin connected to clock pin (SRCLK) of 74HC595
52 #define clockPin 7
53 //Pin connected to data pin (SER) of 74HC595
54 #define dataPin 6
55
56 //Create stopwatch objects
57 Stopwatch SW_secs(Stopwatch::SECONDS);
58 Stopwatch SW_millis; // MILLIS (default)
59
60 void setup() {
61     //Initialize seven segment pins as output from the Arduino
62     for (int i = 0; i < 8; i++) {
63         pinMode(segmentPins[i], OUTPUT);
64     }
65     //Pins for toggling LEDs
66     for (int i=0; i<4; i++) {
67         pinMode(displayPins[i], OUTPUT);
68     }
69     //Turn Off Seven Segment Displays Initially
70     for (int i=0; i<4; i++) {
71         digitalWrite(displayPins[i], LOW);
72     }
73
74     Serial.begin(9600);
75     //Enable internal pullup on button pin
76     pinMode(buttonPin, INPUT_PULLUP);
77     // After setting up the buttons, setup the Bounce instance :
78     debouncer.attach(buttonPin);
79     debouncer.interval(5); // interval in ms
80     //Initialize values
81     num1=0;
82     num2=0;
83     num3=0;
```

```
84     num4=0;
85     //Shift Register Pins
86     pinMode (latchPin, OUTPUT);
87     pinMode (dataPin, OUTPUT);
88     pinMode (clockPin, OUTPUT);
89     //set byte array
90     setDigitVals();
91     //Starting the stopwatch
92     displayValue = 0;
93 }
94
95
96 void loop() {
97     //checking for start/stop
98     // Update the Bounce instance :
99     debouncer.update();
100    //If a fall occurred, the switch was pressed
101    if ( debouncer.fell() ) {
102        buttonPressTimeStamp = millis();
103        if (SW_secs.isRunning()) {
104            //stop if running
105            SW_secs.stop();
106            SW_millis.stop();
107        } else {
108            //start since it was previously stopped
109            SW_secs.start();
110            SW_millis.start();
111        }
112    }
113
114    if ( debouncer.rose() ) {
115        //checking for long button press to reset
116        if (millis() - buttonPressTimeStamp > 2000) {
117            startOver(false);
118        }
119    }
120 }
121
122 //Only incrementing display after every second
123 if ( SW_secs.elapsed() == (displayValue + 1)) {
124     displayValue++;
125     //100 min - 60 x 100
126     if (displayValue == 6000) {
127         //resetting after reaching 100 mins
128         startOver(true);
129     }
130     //get first digit
131     //10's minutes
132     num1 = (displayValue / 60) / 10;
133     //get second digit
134     //minutes
135     num2 = (displayValue / 60) % 10;
```

```
136 //get third digit
137 //10's seconds
138 num3 = (displayValue % 60) / 10;
139 //get fourth digit
140 //seconds
141 num4 = (displayValue % 60) % 10;
142 }
143 //update the seven segment display
144 updateDisplay(num1, num2, num3, num4);
145 }
146
147 void updateDisplay(int digit1, int digit2, int digit3, int digit4) {
148 //Toggle displays on an off and show each digit separately
149 //Show only first digit
150 digitalWrite(displayPins[0], HIGH);
151 digitalWrite(displayPins[1], LOW);
152 digitalWrite(displayPins[2], LOW);
153 digitalWrite(displayPins[3], LOW);
154 setSegments(digit1);
155 delay(2);
156 //Show only second digit
157 digitalWrite(displayPins[0], LOW);
158 digitalWrite(displayPins[1], HIGH);
159 digitalWrite(displayPins[2], LOW);
160 digitalWrite(displayPins[3], LOW);
161 setSegments(digit2);
162 delay(2);
163 //Show only third digit
164 digitalWrite(displayPins[0], LOW);
165 digitalWrite(displayPins[1], LOW);
166 digitalWrite(displayPins[2], HIGH);
167 digitalWrite(displayPins[3], LOW);
168 setSegments(digit3);
169 delay(2);
170 //Show only fourth digit
171 digitalWrite(displayPins[0], LOW);
172 digitalWrite(displayPins[1], LOW);
173 digitalWrite(displayPins[2], LOW);
174 digitalWrite(displayPins[3], HIGH);
175 setSegments(digit4);
176 delay(2);
177 }
178
179 //Todo: Modify to take a boolean to determine if the decimal point should be set
180 void setSegments(int n) {
181 digitalWrite(latchPin, LOW);
182 shiftOut(dataPin, clockPin, LSBFIRST, digitVals[n]);
183 digitalWrite(latchPin, HIGH);
184 }
185
186 //start over the stop watch
187 void startOver(bool startCount) {
```

```
188 //reset stopwatches
189 displayValue = 0;
190 SW_secs.reset();
191 SW_millis.reset();
192 if (startCount) {
193     //automatically restart
194     SW_secs.start();
195     SW_millis.start();
196 } else {
197     //do not restart and set display to 0
198     num1 = 0;
199     num2 = 0;
200     num3 = 0;
201     num4 = 0;
202 }
203 }
204
205 //convert digits bit array into actual array of bytes needed for the shift register
206 void setDigitVals() {
207     for ( int j = 0; j < 10; j++ ) {
208         byte leds = 0;
209         for (int i = 0; i<8; i++) {
210             if (digits[j][i] == 0) {
211                 bitSet(leds, 7-i);
212             } else {
213                 bitClear(leds, 7-i);
214             }
215         }
216         digitVals[j] = leds;
217     }
218 }
```