

Arduino Sketch

/* Steam Powered Clock with LCD display

* Arduino Modules: Uno, XC4450 RTC, LCD 16x2 Keypad Shield, 4-way relay board

*

* Created: 27 March, 2020 by Paul Cooper

* Revisions:

* 4 Aug 2020 - 20 minute time call

* 5 Aug 2020 - A "5 minute prime" added to keep interest going between 20 minute interval performances

* 6 Aug 2020 - Check Mode added to allow manual testing of system

* 9 Aug 2020 - The "5 minute prime" made random for some additional interest

* 10 Aug 2020 - Ability to zero the seconds made to better synchronise time

*

* also see <https://create.arduino.cc/projecthub/electropeak/using-1602-lcd-keypad-shield-w-arduino-w-examples-e02d95>

*

* Clock Adjustment

* -----

* Adjust hours: Press LEFT BUTTON, press UP/DOWN to adjust hour up/down, then press SELECT to set time and return to normal operation

* Adjust Minutes: Press RIGHT BUTTON, press UP/DOWN to adjust minutes up/down, then press SELECT to set time and return to normal operation

* Zero Seconds: Pressing the DOWN BUTTON will zero the seconds to help synchronise time

*

* Check Mode

* -----

* Check mode can be used to ensure wiring is correct, to check the operation of the system and/or to prime the system

* Check Mode: Press UP BUTTON

*

*/

```
#include <Wire.h>
```

```
#include <RTClib.h>
```

```
#include <LiquidCrystal.h>
```

```
int currentHR;
```

```
int currentMIN;
```

```
int wasHR;
```

```
int wasMIN = 12;
```

```
int currentSEC;
```

```
int wasSEC;
```

```
// name the pins that will drive the relays
```

```
const int twelve = 3;
```

```
const int four = 11;
```

```
const int eight = 12;
```

```
const int PuffTime = 175;
```

```
const int PuffWait = 4000;
```

```
//pin for buttons
```

```

#define buttonPRESSED A0
//button constants
#define btnRT 0 // need to test each board as there can be differences.
#define btnUP 99 // example for when testing if/which button pushed....
#define btnDN 257 // if (btnPRESSED > btnRT-50) && btnPRESSED < btnRT+50) {.....
#define btnLT 410
#define btnSET 640
#define btnNONE 1023
//
int btnPRESSED = 0;

RTC_DS1307 rtc;
DateTime now;

// declare the pins used by the LCD panel
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);
// pin 10 is display control - high is ON, low is DIM

void setup() {
  Serial.begin(57600);
  lcd.begin(16, 2); // start the lcd library
  lcd.setCursor(0,0);
  lcd.print("@ the 3rd stroke"); // print welcome message
  lcd.setCursor(0,1);
  lcd.print("...it will be...");
  delay(2000);

  randomSeed(analogRead(1)); //using A1 to randomise
  pinMode(A2, OUTPUT);
  pinMode(A3, OUTPUT);
  analogWrite(A2, 0); // set to GND for RTC module
  analogWrite(A3, 255); // set to 5V for RTC module
  pinMode(four, OUTPUT);
  pinMode(eight, OUTPUT);
  pinMode(twelve, OUTPUT);

  rtc.begin(); // start the RTC library
  delay(250);

  if (! rtc.isrunning()) {
    lcd.setCursor(0,0);
    lcd.print("RTC NOT running!");
    lcd.setCursor(0,1);
    lcd.print(":( ");
    while (1);
  }
  lcd.clear();
}

void loop() {
  lcd.setCursor(0,0);
  lcd.print(" Steam Clock ");
}

```

```

displayTIME();
if (currentMIN != wasMIN) {
  if (currentMIN == 20) twentypast();
  if (currentMIN == 40) twentyto();
  if (currentMIN == 0) o'clock();
  if (currentMIN == 5 || currentMIN == 10) DoAPuff();
  if (currentMIN == 15 || currentMIN == 25) DoAPuff();
  if (currentMIN == 30 || currentMIN == 35) DoAPuff();
  if (currentMIN == 45 || currentMIN == 50) DoAPuff();
  if (currentMIN == 55 ) DoAPuff();
  wasHR = currentHR;
  wasMIN = currentMIN;
  wasSEC = currentSEC;
}
btnPRESSED = analogRead(A0);
if (btnPRESSED > (btnLT-50) && btnPRESSED < (btnLT+50)) {adjustHOURS();} //selection
button
if (btnPRESSED > (btnRT-50) && btnPRESSED < (btnRT+50)) {adjustMINUTES();}
if (btnPRESSED > (btnDN-50) && btnPRESSED < (btnDN+50)) {zeroSECONDS();}
if (btnPRESSED > (btnUP-50) && btnPRESSED < (btnUP+50)) {chkMODE();}
delay(500);
} // end void loop ()

```

```

void displayTIME() {
  DateTime time = rtc.now();
  currentHR = time.hour();
  currentMIN = time.minute();
  currentSEC = time.second();
  lcd.setCursor(4,1);
  if (currentHR <10 ) {lcd.print("0");}
  lcd.print(currentHR);
  lcd.print(':');
  if (currentMIN <10) {lcd.print("0");}
  lcd.print(currentMIN);
  lcd.print(".");
  if (currentSEC <10) {lcd.print("0");}
  lcd.print(currentSEC);
  lcd.print(" ");
}

```

```

void DoAPuff() {
  int ranTime;
  ranTime = random(1,3);
  if (ranTime == 1) {digitalWrite(twelve, HIGH), delay(random(75, PuffTime)),
digitalWrite(twelve, LOW);}
  if (ranTime == 2) {digitalWrite(four, HIGH); delay(random(75, PuffTime)); digitalWrite(four,
LOW);}
  if (ranTime == 3) {digitalWrite(eight, HIGH); delay(random(75, PuffTime)); digitalWrite(eight,
LOW);}
}

```

```

void twentypast() {
  checkHR();
  delay(PuffWait),
  digitalWrite(four, HIGH), delay(PuffTime*1.5), digitalWrite(four, LOW);
}

void twentyto() {
  checkHR();
  delay(PuffWait),
  digitalWrite(eight, HIGH), delay(PuffTime*1.5), digitalWrite(eight, LOW);
}

void oclock() {
  checkHR();
  delay(PuffWait ),
  digitalWrite(twelve, HIGH), delay(PuffTime*1.5), digitalWrite(twelve, LOW);
}

void checkHR() {
  delay(1000);
  if (currentHR == 1 || currentHR == 13) oneoclock();
  if (currentHR == 2 || currentHR == 14) twooclock();
  if (currentHR == 3 || currentHR == 15) fouroclock();
  if (currentHR == 4 || currentHR == 16) fouroclock();
  if (currentHR == 5 || currentHR == 17) fiveoclock();
  if (currentHR == 6 || currentHR == 18) eightoclock();
  if (currentHR == 7 || currentHR == 19) sevenoclock();
  if (currentHR == 8 || currentHR == 20) eightoclock();
  if (currentHR == 9 || currentHR == 21) nineoclock();
  if (currentHR == 10 || currentHR == 22) tenoclock();
  if (currentHR == 11 || currentHR == 23) elevenoclock();
  if (currentHR == 12 || currentHR == 0) twelveoclock();
}

void oneoclock() {
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(four, HIGH); delay(PuffTime); digitalWrite(four, LOW);
}

void twooclock() {
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(four, HIGH); delay(PuffTime); digitalWrite(four, LOW); delay(PuffWait);
  digitalWrite(four, HIGH); delay(PuffTime); digitalWrite(four, LOW);
}

void threeoclock() {
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(four, HIGH); delay(PuffTime); digitalWrite(four, LOW); delay(PuffWait);
  digitalWrite(four, HIGH); delay(PuffTime); digitalWrite(four, LOW); delay(PuffWait);
}

```



```

void elevenoclock() {
  digitalWrite(eight, HIGH); delay(PuffTime); digitalWrite(eight, LOW); delay(PuffWait);
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW);
}

```

```

void twelveoclock() {
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
  digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW);
}

```

```

void adjustHOURS() {
  while (btnPRESSED<2000){
    lcd.setCursor(0,0);
    lcd.print(" Adjust Hours ");
    DateTime time = rtc.now();
    currentHR = time.hour();
    btnPRESSED = analogRead(A0);
    displayTIME();
    if (btnPRESSED > (btnUP-50) && btnPRESSED < btnUP+50) { //UP button
      currentHR = currentHR+1; if(currentHR>23) {currentHR=0;}
    }
    if (btnPRESSED > (btnDN-50) && btnPRESSED < (btnDN+50)) {
      currentHR = currentHR-1; if(currentHR<0) {currentHR=23;}
    }
    rtc.adjust(DateTime(time.year(), time.month(), time.day(), currentHR, time.minute(),
time.second())); //update clock
    displayTIME();
    if (btnPRESSED < (btnRT+50)) {adjustMINUTES();} // RIGHT button
    if (btnPRESSED > (btnSET-50) && btnPRESSED < (btnSET+50)) {return;} // SELECT
button
    btnPRESSED = 1000;
    delay(750);
  }
}

```

```

void adjustMINUTES() {
  while (btnPRESSED<2000){
    lcd.setCursor(0,0);
    lcd.print(" Adjust Minutes");
    DateTime time = rtc.now();
    currentMIN = time.minute();
    btnPRESSED = analogRead(buttonPRESSED);
    displayTIME();
    if (btnPRESSED > (btnUP-50) && btnPRESSED < (btnUP+50)) {
      currentMIN = currentMIN+1; if(currentMIN>59) {currentMIN=0;}
    }
    if (btnPRESSED > (btnDN-50) && btnPRESSED < (btnDN+50)) {
      currentMIN = currentMIN-1; if(currentMIN<0) {currentMIN=59;}
    }
  }
}

```

```

    }
    rtc.adjust(DateTime(time.year(), time.month(), time.day(), time.hour(), currentMIN,
time.second())); //update clock
    displayTIME();
    if (btnPRESSED > (btnLT-50) && btnPRESSED < (btnLT+50)) {adjustHOURS();}
    if (btnPRESSED > (btnSET-50) && btnPRESSED < (btnSET+50)) {return;} // SELECT
button
    btnPRESSED = 1000;
    delay(750);
    }
    }

```

```

void zeroSECONDS() {
    DateTime time = rtc.now();
    rtc.adjust(DateTime(time.year(), time.month(), time.day(), time.hour(), time.minute(), 0));
//update clock
    displayTIME();
}

```

```

void chkMODE() {
    lcd.setCursor(0,0);
    lcd.print("****Check Mode****");
    lcd.setCursor(0,1);
    lcd.print("      ");
    delay(3000);
    digitalWrite(twelve, HIGH); delay(PuffTime); digitalWrite(twelve, LOW); delay(PuffWait);
    digitalWrite(four, HIGH); delay(PuffTime); digitalWrite(four, LOW); delay(PuffWait);
    digitalWrite(eight, HIGH); delay(PuffTime); digitalWrite(eight, LOW);
}

```

```

// END OF SKETCH

```