Countdown to 2011: A Floating LED Clock and Message Display

Maya Varma

Lawson Middle School, Cupertino

Project Number: LED105231

Project Description

The purpose of this project was to design and build a floating LED-based clock and message display using a single row of rotating LEDs. The design consists of a row of 32 RGB LEDs controlled by a microcontroller that is rotated at a speed of around 100 rotations per second. If the LEDs are turned on and off at the appropriate time during each rotation, the eye can "connect the dots" together to form patterns that normally require large two-dimensional arrays of LEDs to display. The patterns can include graphics and text, and can also be animated.

For my project, I chose the New Year theme. The goal was to make the design perform the following sequence of tasks:

1. Start with an analog clock display consisting of the seconds, minutes and hour hands. The initial time should be 11:59:00. When the seconds hand completes one rotation, the minute hand should move to the 12:00 position.
2. At this point, the device should display an animated "ball" dropping down to ring in the new year.
3. The device then should display the text "Happy New Year."
4. Finally, the device should display a flashing sign "2011" .

The design consists of two circuit boards, one a narrow strip for mounting the 32 LEDS in a row, and the other a 4.5x8" main board for all the circuit components. I chose the Arduino Nano 3.0 microcontroller for this project. It takes only a small area on the board and is fast enough to control the 32 LEDs at the required speed. The board has only a small number of outputs, so I used a chain of 8-bit shift registers to drive the LEDs. I constructed three chains of four shift registers each, one for each color. The 32 LEDs can be set to any combination of colors in 32 cycles by shifting data into each chain. This took only three output pins of the Nano to drive the shift register data inputs, another output to clock the data in, and three more outputs to load the outputs of the shift registers after the data has been shifted in (a total of 7 output pins).

I mounted the boards on a floor-standing fan to rotate it. I determined that the fan rotated at about 120 rotations per second in its medium speed setting, which was adequate to create the display patterns. The Arduino could set up the 32 LEDs in less than 100 microseconds, which was adequate for the project. I used the QRD1114 reflective sensor, mounted at the tip of the LED board, to determine the position of the LED arm. I mounted a white piece of paper on the fan cover at the 9 O'clock position to activate the sensor. I used a 9V battery mounted on the board to power the design.

Schematics

(attached separately)

Bill of Materials

Reference number in schematics	Quantity	Part Number	Description	Source
LED1-LED32	32	BL-L515	RGB LED 5mm	Evil Mad Science
U1-U12	12	TPIC6B595	8-bit Shit Register	Sparkfun
U13	1	LM340	5V Regulator	Digi-Key
U14	1	QRD1114	Reflective Sensor	Sparkfun
R1-R96	96		300 ohms, 0.25W Resistor	Digi-Key
R97	1		300 ohms, 0.25W Resistor	Digi-Key
R98	1		10K ohms, 0.25W Resistor	Digi-Key
	16	1109PHCT-ND	0.1 uF 50V ceramic capacitor	Digi-Key
	1	338-1723-ND	220 uF, 25V electrolytic capacitor	Digi-Key
	1		Arduino Nano 3.0 Board	Maker Shed
	1	V1021-ND	Vectorbord 4.5 x 8 inch solder board	Digi-Key
	1		7.5x1.5” solder board (cut from a larger board)	Digi-Key
	12	ED3120-ND	20-pin IC socket	Digi-Key
	1		9V battery	

Source Code

```
/* Rotating LED clock and message display using the Arduino Nano board.

 This code controls single string of 32 RGB LEDs.
 The LEDs are connected to three chains of shift registers, one for each
color.
 The serial data for each of these colors is provided by an Arduino output
pin, one for each color.
 The serial clocks of all the shift registers are connected to another
Arduino output pin.
 The load inputs for each chain of shift registers are also driven by
separate Arduino output pins
 (one for each color).

 The position of the LED string during the rotation is determined using a
light sensor at the 9 O'clock
 position.

 Maya Varma, Dec 2010
 */


// Arduino pin assignments
int redSerialIn = 12;
int greenSerialIn = 11;
int blueSerialIn = 10;
int serialClock = 13;
int redRegLoad = 9;
int greenRegLoad = 8;
int blueRegLoad = 7;
int sensorPin = A0; // Connected to the light detector


long lastSyncTime = 0; // Time at which the last sync signal was detected
from the light sensor (in us)
long currentTime = 0; // current Time from Arduino in us
long rotationTime = 66000; // Time for approx 3/4 rotation of the LEDs

long lastSecondsHandMoveTime = 0;  // Time at which second hand was last
moved
long lastMinutesHandMoveTime = 0;  // Time at which minute hand was last
moved
long lastHoursHandMoveTime = 0;  // Time at which hour hand was last moved

// Current positions of the clock hands from the synchronization point
long secondsHandPosition = 22000; // distance of 12 O'clock position from the
sensor in the 9 O'clock position
long minutesHandPosition = 22000 - 22000/15;
long hoursHandPosition = 22000;

long ballDropPatternPosition = 21000; // Position to display ball drop from
the 9 O'clock position
```

```
long happyNewYearPatternPosition = 10000; // Position to display "Happy New
Year" from the 9 O'clock position

long year2011PatternPosition = 20000; // Position to display "2011" from the
9 O'clock position

// Flags to set hands of the clock
boolean secondsSet = false;
boolean minutesSet = false;
boolean hoursSet = false;
boolean displaySeconds = false;
boolean displayMinutes = false;
boolean displayHours = false;

boolean minuteHandMoved = false;  // Flag to detect when Minutes hand moves
to midnight position

int sensorValue = 0;  // variable to store the value coming from the sensor

int displayMode = 1; // 1 = display clock, 2 = show dropping ball, 3 = show
"Happy New Year" sign,
// 4 = show flashing "2001"

long lastBallMoveTime;  //Time at which the dropping ball was last moved
long happyNewYearStartTime = 0; // Time when "Happy New Year" display starts
long LEDflashTime = 0; // Last time the LED was turned on/off when flashing

// LED patterns for the clock hands
unsigned long secondsPattern = 0xfffffffd;
unsigned long minutesPattern = 0xffffff01;
unsigned long hoursPattern =   0xfffff001;

// LED pattern for dropping ball
unsigned long ballDropPattern[] = {
  0x8, 0x14, 0x22, 0x41, 0xffffffff, 0x41, 0x22, 0x14, 0x8};

// LED pattern for Happy New Year display
unsigned long happyNewYearPattern[] = {
  0x7f, 0x08, 0x08, 0x7f, 0x00, // Letter H
  0x7e, 0x09, 0x09, 0x7e, 0x00, // Letter A
  0x7f, 0x09, 0x09, 0x06, 0x00, // Letter P
  0x7f, 0x09, 0x09, 0x06, 0x00, // Letter P
  0x63, 0x14, 0x08, 0x04, 0x03, 0x00, // Letter Y
  0x00,
  0x7f, 0x06, 0x08, 0x30, 0x7f, 0x00, // Letter N
  0x7f, 0x49, 0x49, 0x41, 0x00, // Letter E
  0x7f, 0x20, 0x10, 0x20, 0x7f, 0x00, // Letter W
  0x00, 0x00,
  0x63, 0x14, 0x08, 0x04, 0x03, 0x00, // Letter Y
  0x7f, 0x49, 0x49, 0x41, 0x00, // Letter E
  0x7e, 0x09, 0x09, 0x7e, 0x00, // Letter A
  0x7f, 0x09, 0x09, 0x76};

// LED pattern for "2011"
unsigned long year2011Pattern[] = {
  0x62, 0x51, 0x49, 0x46, // digit 2
  0x00,
  0x3e, 0x41, 0x3e, // digit 0
  0x00,
  0x42, 0x7f, 0x40, // digit 1
  0x00,
  0x42, 0x7f, 0x40 // digit 1
};

int index = 0;

boolean LEDon = false;
boolean flashLED = false;


void setup() {
  // initialize output pins
  pinMode(redSerialIn, OUTPUT);
  pinMode(greenSerialIn, OUTPUT);
  pinMode(blueSerialIn, OUTPUT);
  pinMode(serialClock, OUTPUT);
  pinMode(redRegLoad, OUTPUT);
  pinMode(greenRegLoad, OUTPUT);
  pinMode(blueRegLoad, OUTPUT);

  digitalWrite(redSerialIn, LOW);
  digitalWrite(greenSerialIn, LOW);
  digitalWrite(blueSerialIn, LOW);
  digitalWrite(serialClock, LOW);
  digitalWrite(redRegLoad, LOW);
  digitalWrite(greenRegLoad, LOW);
  digitalWrite(blueRegLoad, LOW);
}

void loop() {
  int i;
  unsigned long currentTime = micros();

  // Synchronize the rotation of the LEDs.
  // Each time the sensor output is detected, wait for about 3/4 of the
rotation, then look for the
  // sensor output again.

  if (currentTime - lastSyncTime >= rotationTime){
    // read the value from the sensor:
    sensorValue = analogRead(sensorPin);
    if(sensorValue < 800){
      lastSyncTime = currentTime;
      secondsSet = false;
      minutesSet = false;
      hoursSet = false;
      index = 0;
    }
  }

  switch(displayMode){
  case 1: // Show clock
```

```
    // Check if it is time to display the clock hands.
    // If it is, set the corresponding flag

    if ((currentTime - lastSyncTime >= secondsHandPosition) &&
      !secondsSet){
      secondsSet = true;
      displaySeconds = true;
    }
    else
      displaySeconds = false;

    if ((currentTime - lastSyncTime >= minutesHandPosition) &&
      !minutesSet){
      minutesSet = true;
      displayMinutes = true;
    }
    else
      displayMinutes = false;

    if ((currentTime - lastSyncTime >= hoursHandPosition) &&
      !hoursSet){
      hoursSet = true;
      displayHours = true;
    }
    else
      displayHours = false;

    displayClockHands(displaySeconds, displayMinutes, displayHours);

    // Check if it is time to move the hands
    if (currentTime-lastSecondsHandMoveTime >= 1000000){ // Move every second
      secondsHandPosition = secondsHandPosition + 22000/15;
      lastSecondsHandMoveTime = currentTime;
      if (secondsHandPosition > 87000)
        secondsHandPosition = secondsHandPosition - 87000;
    }

    if (currentTime-lastMinutesHandMoveTime >= (59*1000000)){ // Move every
60 seconds
      minutesHandPosition = minutesHandPosition + 22000/15;
      lastMinutesHandMoveTime = currentTime;
      if (minutesHandPosition > 87000)
        minutesHandPosition = minutesHandPosition - 87000;
      minuteHandMoved = true;
    }

    if (currentTime-lastHoursHandMoveTime >= (60*1000000)){ // Move every 60
seconds
      hoursHandPosition = hoursHandPosition + 22000/(15*60);
      lastHoursHandMoveTime = currentTime;
      if (hoursHandPosition >= 87000)
        hoursHandPosition = hoursHandPosition - 87000;
    }

    // Switch to displaying dropping ball 0.5 sec after minute hand moved

    if (minuteHandMoved && (currentTime-lastSecondsHandMoveTime >= 500000)){
      displayMode = 2;
      lastBallMoveTime = currentTime;
    }
    break;

  case 2: // Show dropping ball
    if (currentTime - lastSyncTime >= ballDropPatternPosition){
      if (index < 9){
        setGreenLED(ballDropPattern[index]);
        index = index+1;
        LEDon = true;
      }
      else{
        setGreenLED(0);
        LEDon = false;
      }
    }
    else
      if (LEDon)
        setGreenLED(0);

    // Move ball down by 1 position after each 0.5 second

    if (currentTime-lastBallMoveTime >= 500000){
      lastBallMoveTime = currentTime;
      for (i=0; i < 9; i++)
        ballDropPattern[i] = ballDropPattern[i] << 1;
      ballDropPattern[4] = 0xffffffff;
      if (ballDropPattern[0] == 0x00800000){
        happyNewYearStartTime = currentTime;
        displayMode = 3;
      }
    }
    break;

  case 3:
    // Display "Happy New Year"
    if (currentTime - lastSyncTime >= happyNewYearPatternPosition){
      if (index < 66){
        setGreenLED(happyNewYearPattern[index]);
        index = index+1;
        LEDon = true;
      }
      else{
        setGreenLED(0);
        LEDon = false;
      }
    }
    else
      if (LEDon)
        setGreenLED(0);
    if (currentTime-happyNewYearStartTime >= 2000000){ // Stop after 2
seconds
      setGreenLED(0);
      displayMode = 4;
      flashLED = true;
      LEDflashTime = currentTime;
    }
    break;

  case 4:
    // Display flashing 2011 sign
    if (currentTime - lastSyncTime >= year2011PatternPosition){
      if (!flashLED)
        setGreenLED(0);
      else
        if (index < 16){
          setGreenLED(year2011Pattern[index]);
          index = index+1;
          LEDon = true;
        }
      else{
        setGreenLED(0);
        LEDon = false;
      }
    }
    else
      if (LEDon)
        setGreenLED(0);
    if (currentTime-LEDflashTime >= 500000){ // Turn off and on after 0.5
second
      LEDflashTime = currentTime;
      flashLED = !flashLED;
      index = 16;
    }
    break;

  default:
    setGreenLED(0);
  }

}

// Display the clock hands at their current positions
void displayClockHands(boolean secondsSet,
boolean minutesSet, boolean hoursSet){

  // Turn on the LEDs for the clock hands

  if (secondsSet)
    setGreenLED(secondsPattern);
  else if (minutesSet)
    setGreenLED(minutesPattern);
  else if (hoursSet)
    setGreenLED(hoursPattern);
  else
    setGreenLED(1);
}

/* The following Function sets up the 32 Red LEDs from a 32-bit input */
void setRedLED (unsigned long x){
  int i;

  for (i=0; i<32; i++){
    // Set up input of shift register chain
    if (x & 1)
      digitalWrite(redSerialIn, HIGH);
    else
      digitalWrite(redSerialIn, LOW);
    // Pulse the serial clock
    digitalWrite(serialClock, HIGH); // serial clock
    digitalWrite(serialClock, LOW);
    x = x >> 1;
  }
  // Now load the outputs of the shift register by giving a load pulse
  digitalWrite(redRegLoad, HIGH);
  digitalWrite(redRegLoad, LOW);
}

/* The following Function sets up the 32 Green LEDs from a 32-bit input */
void setGreenLED (unsigned long x){
  int i;

  for (i=0; i<32; i++){
    // Set up input of shift register chain
    if (x & 1)
      digitalWrite(greenSerialIn, HIGH);
    else
      digitalWrite(greenSerialIn, LOW);
    // Pulse the serial clock
    digitalWrite(serialClock, HIGH); // serial clock
    digitalWrite(serialClock, LOW);
    x = x >> 1;
  }
  // Now load the outputs of the shift register by giving a load pulse
  digitalWrite(greenRegLoad, HIGH);
  digitalWrite(greenRegLoad, LOW);
}

/* The following Function sets up the 32 Blue LEDs from a 32-bit input */
void setBlueLED (unsigned long x){
  int i;

  for (i=0; i<32; i++){
    // Set up input of shift register chain
    if (x & 1)
      digitalWrite(blueSerialIn, HIGH);
    else
      digitalWrite(blueSerialIn, LOW);
    // Pulse the serial clock
    digitalWrite(serialClock, HIGH); // serial clock
    digitalWrite(serialClock, LOW);
    x = x >> 1;
  }
  // Now load the outputs of the shift register by giving a load pulse
  digitalWrite(blueRegLoad, HIGH);
  digitalWrite(blueRegLoad, LOW);
}
```

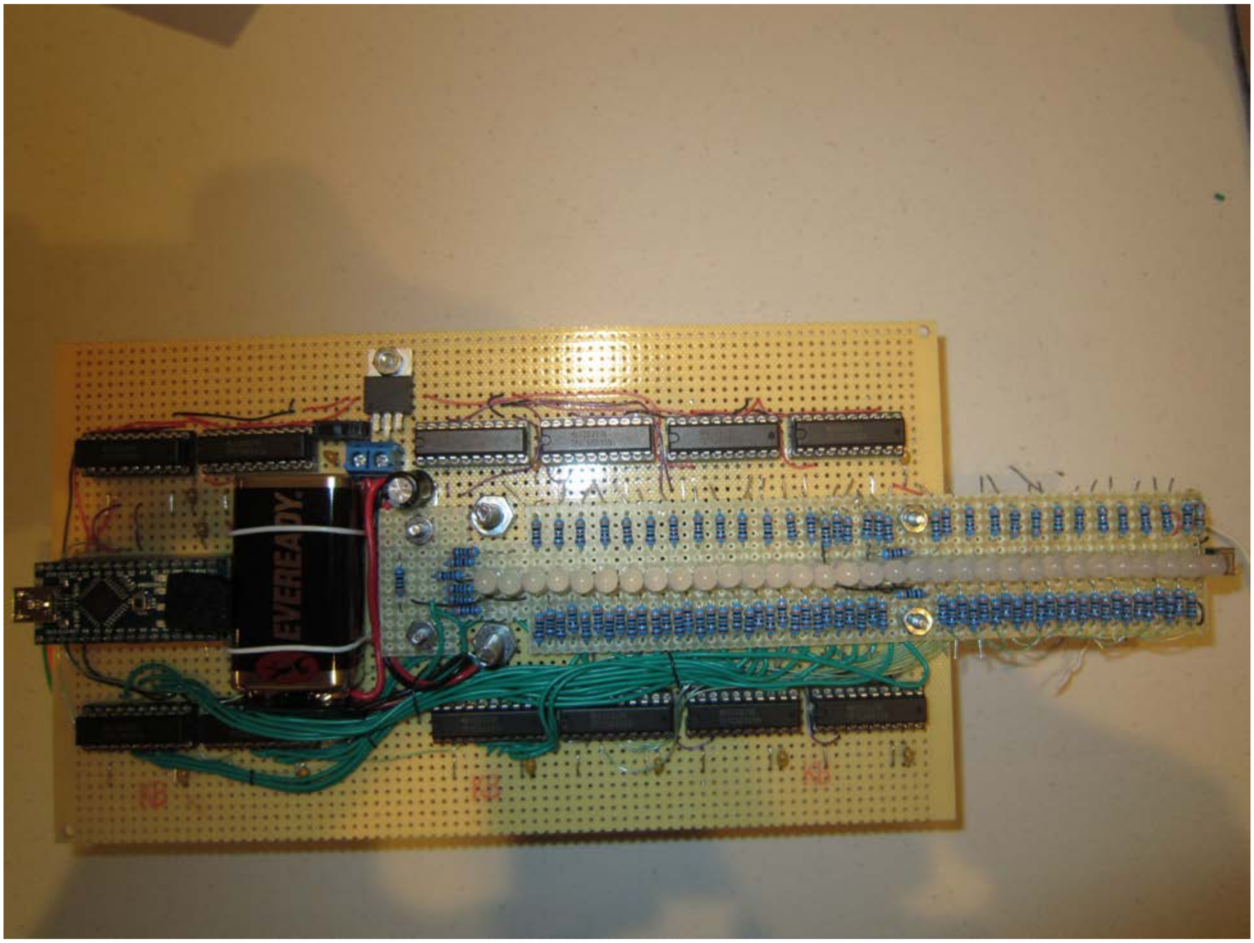
Images

A video on the project has been posted on YouTube at

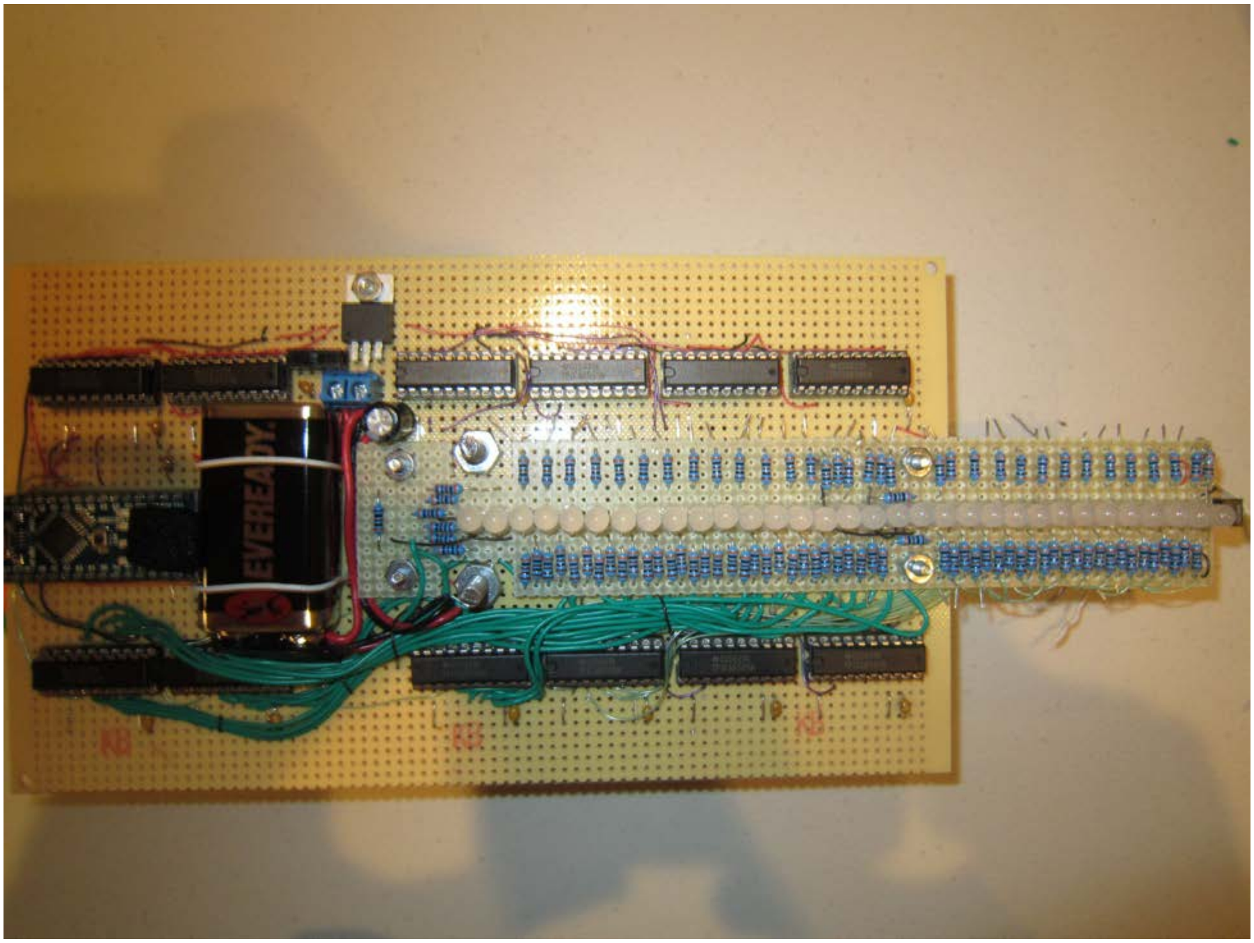
http://www.youtube.com/watch?v=zxIRyXMSpr8

This can also be accessed using the shortened URL

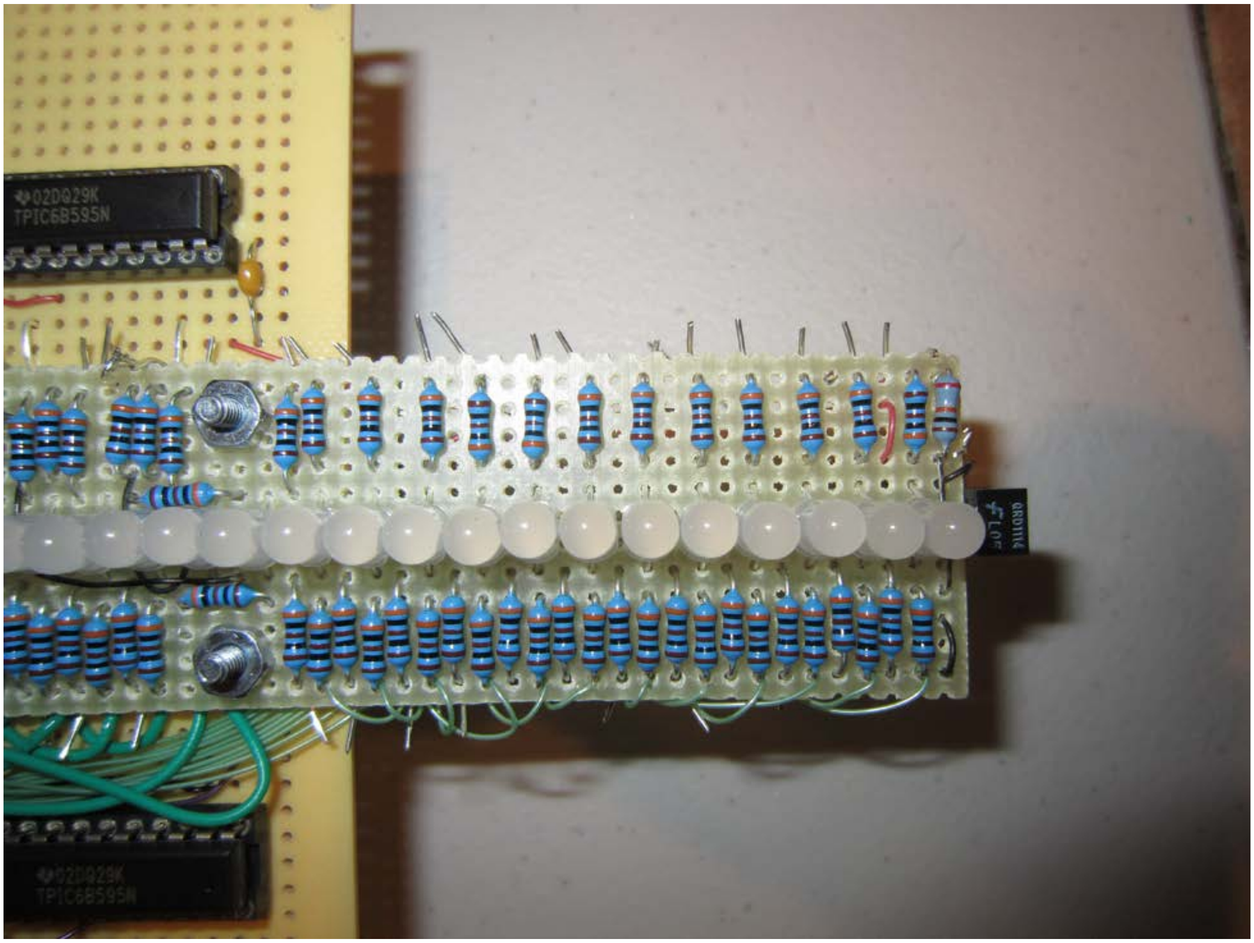
http://bit.ly/hEatYT



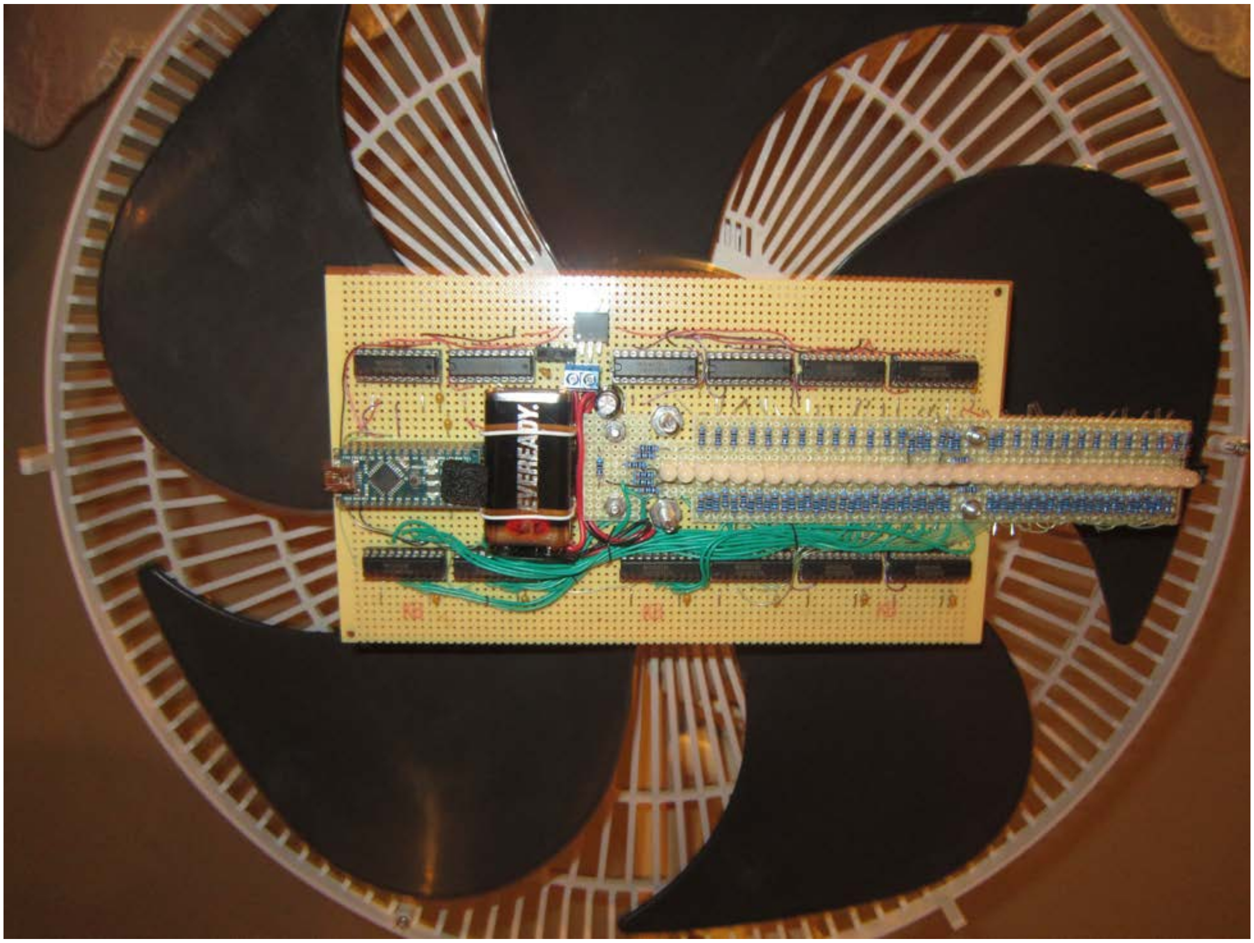
Main board with the LED board mounted on it.



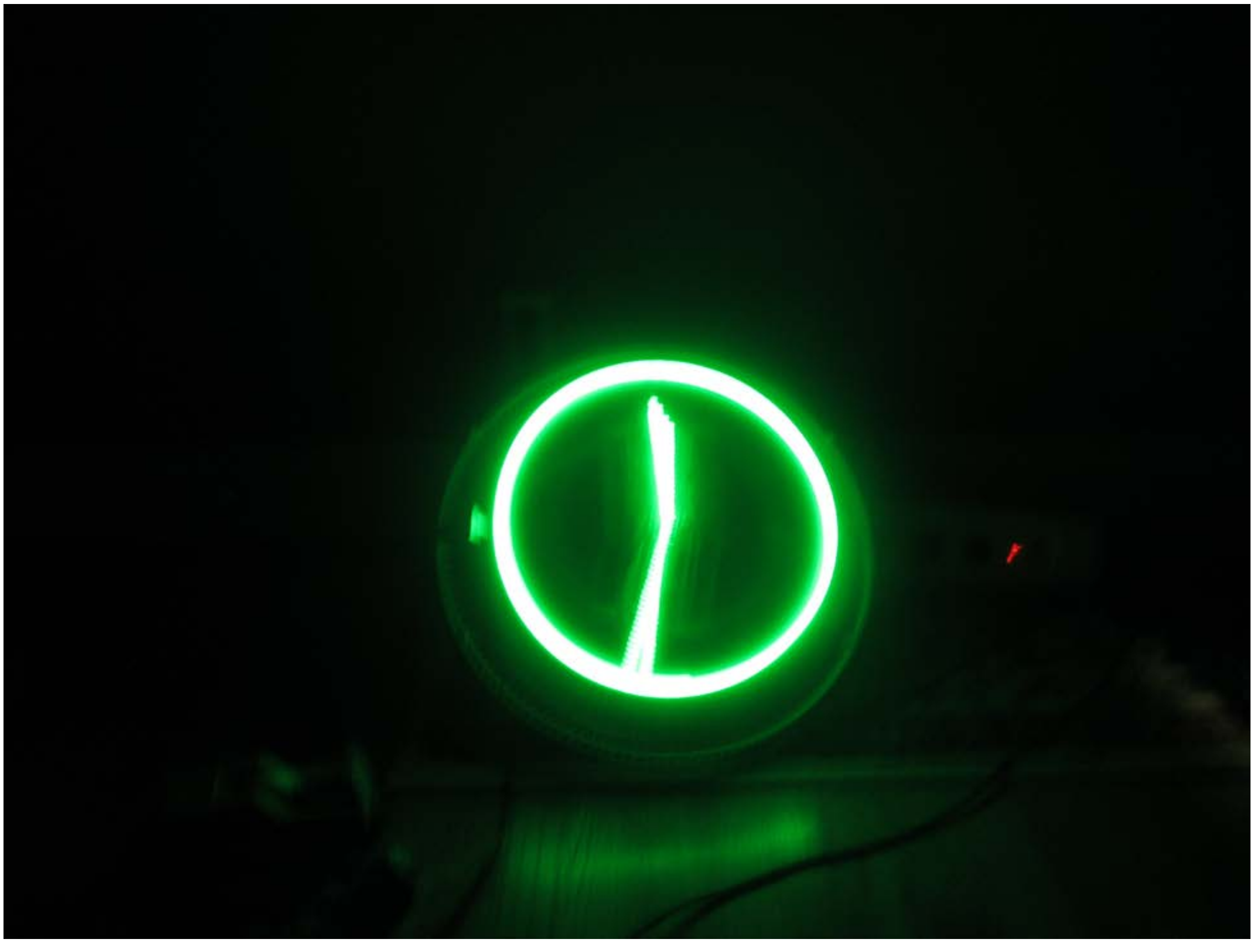
LED board showing the 32 RGB LEDs.



LED board showing the QRD1114 sensor mounted on its edge.



Boards mounted on the fan.



Clock display created by the rotating LEDs.

Happy New Year sign generated by the LEDs.