/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Paws to Wash Handwashing TimerPeter Simpson, EPAM-Continuum March 2020Prox sensor VCNL4010 senses hand near soap dispenser, begins 30 sec NeoPixel countdown and plays Jeopardy timer tune.After completion, there's a 5 sec delay before it can be retriggered.DFRobot MP3 player needs a micro-SD card with /mp3/0001.mp3 tune file.\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/#include "Arduino.h"#include "SoftwareSerial.h"#include "DFRobotDFPlayerMini.h"#include <Wire.h>#include "Adafruit\_VCNL4010.h"#include <Adafruit\_NeoPixel.h>Adafruit\_VCNL4010 vcnl;int prox = 0; // read proximity value ~2200, higher is closerint activate = 0; // indicates timer has been activatedint LED = 13; // trigger enabled diag LED (now onboard)int trigEnbl = 0; // timer can only be triggered when this is 1const unsigned PROX\_LIMIT = 2400; // value above this will trigger timerconst byte BRIGHTNESS = 64; // out of 255 const unsigned long TRIG\_DLY\_TIME = 5000; // 5sec retrigger delayconst unsigned long WASH\_TIME = 30500; // 30.5 second tune timeconst unsigned long NEOPIXEL\_TIME = 1120; // 30.5/24 tweaked so last light goes out before tune ends/\* changed from #defined constants to variables 2015-12-15 Mel Lester Jr. The following two lines are all you may need to change for your project \*/int dataPin = 6; // Arduino PWM data pin D6 int pixels = 24; // number of Neopixelsint i; // pixel counterint playing = 0; // active// Parameter 1 = number of pixels in strip// Parameter 2 = Arduino pin number (most are valid)// Parameter 3 = pixel type flags, add together as needed:// NEO\_KHZ800 800 KHz bitstream (most NeoPixel products w/WS2812 LEDs)// NEO\_KHZ400 400 KHz (classic 'v1' (not v2) FLORA pixels, WS2811 drivers)// NEO\_GRB Pixels are wired for GRB bitstream (most NeoPixel products)// NEO\_RGB Pixels are wired for RGB bitstream (v1 FLORA pixels, not v2)Adafruit\_NeoPixel strip = Adafruit\_NeoPixel(pixels, dataPin, NEO\_GRB + NEO\_KHZ800);uint32\_t Wheel(byte WheelPos);void colorWipe(uint32\_t c, uint8\_t wait);// IMPORTANT: To reduce NeoPixel burnout risk, add 1000 uF capacitor across// pixel power leads, add 300 - 500 Ohm resistor on first pixel's data input// and minimize distance between Arduino and first pixel. Avoid connecting// on a live circuit...if you must, connect GND first.static unsigned long timerTrigDly = millis();SoftwareSerial mySoftwareSerial(10, 11); // RX, TXDFRobotDFPlayerMini myDFPlayer;void printDetail(uint8\_t type, int value);void setup(){ mySoftwareSerial.begin(9600); Serial.begin(9600);//NeoPixel init strip.begin(); strip.show(); // Initialize all pixels to 'off' // Prox sensor init // Prox detect LED pinMode(LED, OUTPUT); Serial.println("VCNL4010 test"); if (! vcnl.begin()){ Serial.println("Sensor not found :("); while (1); } Serial.println("Found VCNL4010");// DFPlayer init Serial.println(); Serial.println(F("DFRobot DFPlayer Mini Demo")); Serial.println(F("Initializing DFPlayer ... (May take 3~5 seconds)")); if (!myDFPlayer.begin(mySoftwareSerial)) { //Use softwareSerial to communicate with mp3. Serial.println(F("Unable to begin:")); Serial.println(F("1.Please recheck the connection!")); Serial.println(F("2.Please insert the SD card!")); while(true); } Serial.println(F("DFPlayer Mini online."));// myDFPlayer.setTimeOut(20000); //Set serial communictaion time out 500ms //----Set volume---- myDFPlayer.volume(10); //Set volume value (0~30). //----Set different EQ---- myDFPlayer.EQ(DFPLAYER\_EQ\_NORMAL); //----Set device we use SD as default---- myDFPlayer.outputDevice(DFPLAYER\_DEVICE\_SD); //----Mp3 control----// myDFPlayer.sleep(); //sleep// myDFPlayer.reset(); //Reset the module// myDFPlayer.enableDAC(); //Enable On-chip DAC// myDFPlayer.disableDAC(); //Disable On-chip DAC// myDFPlayer.outputSetting(true, 15); //output setting, enable the output and set the gain to 15 //----Mp3 play---- //myDFPlayer.play(1); //Play the first mp3 //----Read imformation---- Serial.println("MP3 State: "); Serial.println(myDFPlayer.readState()); //read mp3 state Serial.println("Volume: "); Serial.println(myDFPlayer.readVolume()); //read current volume Serial.println("EQ: "); Serial.println(myDFPlayer.readEQ()); //read EQ setting Serial.println("File Count: "); Serial.println(myDFPlayer.readFileCounts()); //read all file counts in SD card Serial.println("Now Playing: "); Serial.println(myDFPlayer.readCurrentFileNumber()); //read current play file number Serial.println("MP3 files: "); Serial.println(myDFPlayer.readFileCountsInFolder(3)); //read fill counts in folder SD:/03// Activate initial trigger delay timer timerTrigDly = millis();}void loop(){ static unsigned long timerWashTime; static unsigned long timerNeoPixel; static unsigned long timer; // If trigger delay timer timed out if ( (millis() - timerTrigDly > TRIG\_DLY\_TIME) && playing==0 ) { trigEnbl = 1; digitalWrite(LED, 1); Serial.println("Ready"); }// Check the prox sensor prox = vcnl.readProximity(); delay(100); // If sensor active if ((prox > PROX\_LIMIT) && (trigEnbl)) { trigEnbl = 0; activate = 1; Serial.println("Triggered"); } if (activate) { //turn on neopixel colorWipe(strip.Color(0, BRIGHTNESS, 0), 25); // Start playing the tune myDFPlayer.volume(10); //Set volume value (0~30). delay(500); myDFPlayer.play(1); //Play the first mp3 //trigger neopixel timer i = pixels; timerNeoPixel = millis(); // Start the 20 sec countdown timer & NeoPixel Timer timerWashTime = millis(); // end of activation segment playing = 1; activate = 0; Serial.println("Playing"); }// If wash timer still active if (millis() - timerWashTime < WASH\_TIME) {// if Neopixel timer expired if (millis() - timerNeoPixel > NEOPIXEL\_TIME) {// set next Neopixel mode (and turn off trigger LED) digitalWrite(LED, 0); strip.setPixelColor(i, 0); strip.show(); i--;// reset neopixel timer timerNeoPixel = millis(); } } // If wash timer expired else if (playing){ // kill music myDFPlayer.reset(); playing = 0; // blank neopixels digitalWrite(LED, 0); //Reset trigger delay timer timerTrigDly = millis(); Serial.println("Done"); } if (playing){ if (myDFPlayer.available()) { printDetail(myDFPlayer.readType(), myDFPlayer.read()); //Print the detail message from DFPlayer to handle different errors and states. } } }// Fill the dots one after the other with a colorvoid colorWipe(uint32\_t c, uint8\_t wait) { for(uint16\_t i=0; i<strip.numPixels(); i++) { strip.setPixelColor(i, c); strip.show(); delay(wait); }}// Input a value 0 to 255 to get a color value.// The colours are a transition r - g - b - back to r.uint32\_t Wheel(byte WheelPos) { WheelPos = 255 - WheelPos; if(WheelPos < 85) { return strip.Color(255 - WheelPos \* 3, 0, WheelPos \* 3); } else if(WheelPos < 170) { WheelPos -= 85; return strip.Color(0, WheelPos \* 3, 255 - WheelPos \* 3); } else { WheelPos -= 170; return strip.Color(WheelPos \* 3, 255 - WheelPos \* 3, 0); }}void printDetail(uint8\_t type, int value){ switch (type) { case TimeOut: Serial.println(F("Time Out!")); break; case WrongStack: Serial.println(F("Stack Wrong!")); break; case DFPlayerCardInserted: Serial.println(F("Card Inserted!")); break; case DFPlayerCardRemoved: Serial.println(F("Card Removed!")); break; case DFPlayerCardOnline: Serial.println(F("Card Online!")); break; case DFPlayerPlayFinished: Serial.print(F("Number:")); Serial.print(value); Serial.println(F(" Play Finished!")); break; case DFPlayerError: Serial.print(F("DFPlayerError:")); switch (value) { case Busy: Serial.println(F("Card not found")); break; case Sleeping: Serial.println(F("Sleeping")); break; case SerialWrongStack: Serial.println(F("Get Wrong Stack")); break; case CheckSumNotMatch: Serial.println(F("Check Sum Not Match")); break; case FileIndexOut: Serial.println(F("File Index Out of Bound")); break; case FileMismatch: Serial.println(F("Cannot Find File")); break; case Advertise: Serial.println(F("In Advertise")); break; default: break; } break; default: break; }}