#include <NewPing.h>

#define SONAR\_NUM 8 // Number of sensors

#define MAX\_DISTANCE 100 // Maximum distance (in cm) to ping

#define PING\_INTERVAL 35 // Time between pings for each sensor

int dis = 15;

int hepr = 3;

int hepl = 5;

int pir = 2;

// LED states

int led1State = LOW, led2State = LOW, led3State = LOW, led4State = LOW;

int led5State = LOW, led6State = LOW, led7State = LOW, led8State = LOW;

int period = 5000; // LED on-time in ms

unsigned long time\_now = 0;

unsigned long pingTimer[SONAR\_NUM];

unsigned int cm[SONAR\_NUM]; // Raw sensor distances

unsigned int sensors[SONAR\_NUM]; // Filtered sensor distances

uint8\_t currentSensor = 0;

double P[SONAR\_NUM] = {0};

double U\_hat[SONAR\_NUM] = {0};

// Kalman Filter per sensor

double kalman(double U, uint8\_t index) {

const double R = 40;

const double H = 1.0;

const double Q = 10;

double K = P[index] \* H / (H \* P[index] \* H + R);

U\_hat[index] = U\_hat[index] + K \* (U - H \* U\_hat[index]);

P[index] = (1 - K \* H) \* P[index] + Q;

return U\_hat[index];

}

// Define the sensor pins

NewPing sonar[SONAR\_NUM] = {

NewPing(8, 8, MAX\_DISTANCE),

NewPing(9, 9, MAX\_DISTANCE),

NewPing(10, 10, MAX\_DISTANCE),

NewPing(11, 11, MAX\_DISTANCE),

NewPing(12, 12, MAX\_DISTANCE),

NewPing(7, 7, MAX\_DISTANCE),

NewPing(6, 6, MAX\_DISTANCE),

NewPing(4, 4, MAX\_DISTANCE),

};

void setup() {

DDRD |= B11111000; // Set digital pins 3-7 as outputs (used for PWM)

pinMode(pir, INPUT);

pinMode(hepl, OUTPUT);

pinMode(hepr, OUTPUT);

Serial.begin(9600);

pingTimer[0] = millis() + 75;

for (uint8\_t i = 1; i < SONAR\_NUM; i++)

pingTimer[i] = pingTimer[i - 1] + PING\_INTERVAL;

}

void loop() {

for (uint8\_t i = 0; i < SONAR\_NUM; i++) {

if (millis() >= pingTimer[i]) {

pingTimer[i] += PING\_INTERVAL \* SONAR\_NUM;

if (i == 0 && currentSensor == SONAR\_NUM - 1) {

oneSensorCycle();

led();

}

if ((millis() >= time\_now + period) && (

led1State == HIGH || led2State == HIGH || led3State == HIGH ||

led4State == HIGH || led5State == HIGH)) {

time\_now = millis();

led1State = led2State = led3State = led4State = led5State = LOW;

led6State = led7State = led8State = LOW;

}

sonar[currentSensor].timer\_stop();

currentSensor = i;

cm[currentSensor] = 0;

sonar[currentSensor].ping\_timer(echoCheck);

}

}

}

void echoCheck() {

if (sonar[currentSensor].check\_timer())

cm[currentSensor] = sonar[currentSensor].ping\_result / US\_ROUNDTRIP\_CM;

}

void oneSensorCycle() {

for (uint8\_t i = 0; i < SONAR\_NUM; i++) {

sensors[i] = kalman(cm[i], i);

Serial.print("Sensor ");

Serial.print(i);

Serial.print(" Distance: ");

Serial.println(sensors[i]);

}

// LED logic

if (sensors[0] > 0 && sensors[0] <= dis) led1State = HIGH;

if (sensors[1] > 0 && sensors[1] <= dis) led2State = HIGH;

if (sensors[2] > 0 && sensors[2] <= dis) led3State = HIGH;

if (sensors[3] > 0 && sensors[3] <= dis) led4State = HIGH;

if (sensors[4] > 0 && sensors[4] <= dis) led5State = HIGH;

if (sensors[5] > 0 && sensors[5] <= dis) led6State = HIGH;

if (sensors[6] > 0 && sensors[6] <= dis) led7State = HIGH;

if (sensors[7] > 0 && sensors[7] <= dis) led8State = HIGH;

if (sensors[0] <= dis && sensors[1] <= dis) led1State = led2State = HIGH;

if (sensors[1] <= dis && sensors[2] <= dis) led2State = led3State = HIGH;

if (sensors[2] <= dis && sensors[3] <= dis) led3State = led4State = HIGH;

if (sensors[3] <= dis && sensors[4] <= dis) led4State = led5State = HIGH;

if (sensors[4] <= dis && sensors[5] <= dis) led5State = led6State = HIGH;

if (sensors[5] <= dis && sensors[6] <= dis) led6State = led7State = HIGH;

if (sensors[6] <= dis && sensors[7] <= dis) led7State = led8State = HIGH;

if (sensors[7] <= dis && sensors[0] <= dis) led8State = led1State = HIGH;

}

void led() {

if (led1State == HIGH) {

analogWrite(hepl, 200); delay(700); analogWrite(hepl, 0);

}

if (led2State == HIGH) {

analogWrite(hepr, 200); delay(700); analogWrite(hepr, 0);

}

if (led3State == HIGH) {

analogWrite(hepl, 200); delay(2500); analogWrite(hepl, 0);

}

if (led4State == HIGH) {

analogWrite(hepr, 200); delay(2500); analogWrite(hepr, 0);

}

if (led5State == HIGH) {

for (int i = 0; i < 3; i++) {

analogWrite(hepl, 200); delay(400); analogWrite(hepl, 0); delay(300);

}

}

if (led6State == HIGH) {

for (int i = 0; i < 3; i++) {

analogWrite(hepr, 200); delay(400); analogWrite(hepr, 0); delay(300);

}

}

if (led7State == HIGH) {

analogWrite(hepl, 100); delay(1000); analogWrite(hepl, 0);

}

if (led8State == HIGH) {

analogWrite(hepr, 100); delay(1000); analogWrite(hepr, 0);

}

if (digitalRead(pir) == HIGH) {

analogWrite(hepl, 200);

analogWrite(hepr, 200);

delay(3000);

analogWrite(hepl, 0);

analogWrite(hepr, 0);

}

}