#include <TMRpcm.h>

#include <SD.h> // need to include the SD library

#define SD\_ChipSelectPin 53 //using digital pin 4 on arduino nano 328 // also need to include this library...

TMRpcm tmrpcm; // create an object for use in this sketch

int cs,rs,a1,a2,a3,PB=0,s,distance,duration,temp=0;

int x;

int flag=0;

void setup()

{

Serial.begin(9600);

pinMode(A3,INPUT);

pinMode(A4,INPUT);

pinMode(A5,INPUT);

pinMode(A1,INPUT);

pinMode(A2,INPUT);

pinMode(6,OUTPUT);

pinMode(7,OUTPUT);

pinMode(8,OUTPUT);

pinMode(9,OUTPUT);

pinMode(3,INPUT); // Push button

pinMode(12,INPUT); // Ultrasonic sensor

pinMode(10,OUTPUT);

tmrpcm.speakerPin = 11; //11 on Mega, 9 on Uno, Nano, etc

if (!SD.begin(SD\_ChipSelectPin))

{

return; // don't do anything more if not

}

tmrpcm.volume(1);

}

void front()

{

digitalWrite(6,HIGH);

digitalWrite(7,LOW);

digitalWrite(8,HIGH);

digitalWrite(9,LOW);

}

void right()

{

digitalWrite(6,LOW);

digitalWrite(7,LOW);

digitalWrite(8,HIGH);

digitalWrite(9,LOW);

}

void reset()

{

digitalWrite(6,LOW);

digitalWrite(7,LOW);

digitalWrite(8,LOW);

digitalWrite(9,LOW);

}

void order()

{

PB=0;

{

temp=1;

if((cs==0) && (rs==1))

{

front();

if((a1==1)&&(a2==0)&&(a3==1))

{

reset();

if(temp==1)

{

tmrpcm.play("4.wav");

}

}

else if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

else if((cs==1) && (rs==0))

{

right();

if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

if(flag==0)

{

Serial.println("Kitchen");

flag=1;

}

}

}

void loop() {

cs=digitalRead(A1);

rs=digitalRead(A2);

a1=digitalRead(A3);

a2=digitalRead(A4);

a3=digitalRead(A5);

s=digitalRead(3);

digitalWrite(10,HIGH);

delay(10);

digitalWrite(10,LOW);

duration=pulseIn(12,HIGH);

distance=(duration/2)/29.1;

if(s==1)

{

PB=1;

}

if(Serial.available() > 0)

{

x=Serial.read();

flag=0;

}

if(x=='1')

{

temp=1;

if((cs==0) && (rs==1))

{

front();

if((a1==0)&&(a2==0)&&(a3==1))

{

reset();

if(temp==1)

{

tmrpcm.play("1.wav");

}

if(PB==1)

{

order();

}

}

else if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

else if((cs==1) && (rs==0))

{

right();

if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

if(flag==0)

{

Serial.println("table 1");

flag=1;

}

}

else if(x=='2')

{

temp=1;

if((cs==0) && (rs==1))

{

front();

if((a1==0)&&(a2==1)&&(a3==0))

{

reset();

if(temp==1)

{

tmrpcm.play("1.wav");

}

if(PB==1)

{

order();

}

}

else if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

else if((cs==1) && (rs==0))

{

right();

if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

if(flag==0)

{

Serial.println("table 2");

flag=1;

}

}

else if(x=='3')

{

temp=1;

if((cs==0) && (rs==1))

{

front();

if((a1==0)&&(a2==1)&&(a3==1))

{

reset();

if(temp==1)

{

tmrpcm.play("1.wav");

}

if(PB==1)

{

order();

}

}

else if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

else if((cs==1) && (rs==0))

{

right();

if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

if(flag==0)

{

Serial.println("table 3");

flag=1;

}

}

else if(x=='4')

{

temp=1;

if((cs==0) && (rs==1))

{

front();

if((a1==1)&&(a2==0)&&(a3==0))

{

reset();

if(temp==1)

{

tmrpcm.play("1.wav");

}

if(PB==1)

{

order();

}

}

else if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

else if((cs==1) && (rs==0))

{

right();

if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

if(flag==0)

{

Serial.println("table 4");

flag=1;

}

}

else if(x=='5')

{

temp=1;

if((cs==0) && (rs==1))

{

front();

if((a1==1)&&(a2==0)&&(a3==1))

{

reset();

if(temp==1)

{

tmrpcm.play("4.wav");

}

}

else if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

else if((cs==1) && (rs==0))

{

right();

if(distance<15)

{

reset();

if(temp==1)

{

tmrpcm.play("2.wav");

}

}

}

if(flag==0)

{

Serial.println("Kitchen");

flag=1;

}

}

else

{

reset();

if(flag==0)

{

Serial.println("no operation");

flag=1;

}

}

}