/\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @file vr\_sample\_multi\_cmd.ino

\* @brief This file provides a demostration on

how to implement a multi voice command project (exceed 7 voice command)

by using VoiceRecognitionModule

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @note:

voice control led

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* @section HISTORY

\*/

#include <SoftwareSerial.h>

#include "VoiceRecognitionV3.h"

#include <Servo.h>

/\*\*

Connection

Arduino VoiceRecognitionModule

2 -------> TX

3 -------> RX

\*/

VR myVR(2,3); // 2:RX 3:TX, you can choose your favourite pins.

uint8\_t record[7]; // save record

uint8\_t buf[64];

int pos=0;

int pos1=0;

int pos2=0;

Servo myservo1; //door

Servo myservo2; //curtains right

Servo myservo3; //curtains left

int dcm1 = 7; //fan

int dcm2 = 8; //sprinkler

int led1 = 9; //kitchen

int led2 = 6; //living room

int led3 = 11; //bedroom

int le = 6;

int buzz1 = 12; //doorbell

int buzz2 = 13; //burglar

int led=4; //indication

int group = 0;

int brightness = 0; // how bright the LED is

int fadeAmount = 50; // how many points to fade the LED by

//group - 0 (base)

#define oneRecord (25) //access

#define twoRecord (26) //lights

#define threeRecord (27) //fans

#define fourRecord (28) //utilities

#define fiveRecord (29) //safety

#define sixRecord (30)

#define sevenRecord (31)

//group - 1 ACCESS (door,doorbell)

#define group1Record1 (1) //door on

#define group1Record2 (2) //door off

#define group1Record3 (3) //doorbell on

#define group1Record4 (4) //doorbell off

#define group1Record5 (5)

#define group1Record6 (6)

//group - 2 LIGHTS (lights-kitchen,living room,bedroom)

#define group2Record1 (7) //kitchen on

#define group2Record2 (8) //k off

#define group2Record3 (9) //hall bright

#define group2Record4 (10) //h off

#define group2Record5 (11) //room off

#define group2Record6 (12) //r off

//group - 3 FANS

#define group3Record1 (13) //room fan

#define group3Record2 (14) // fan off

#define group3Record3 (15)

#define group3Record4 (16)

#define group3Record5 (17)

#define group3Record6 (18)

//group - 4 UTILITIES

#define group4Record1 (19) //curtain on

#define group4Record2 (20) //curt off

#define group4Record3 (21) //sprinkler

#define group4Record4 (22) //sprink off

#define group4Record5 (23)

#define group4Record6 (24)

//group - 5 SAFETY

#define group5Record1 (32) //help (siren)

#define group5Record2 (33) //silence

#define group5Record3 (34) //alarm off

#define group5Record4 (35)

#define group5Record5 (36)

#define group5Record6 (37)

void setup()

{

/\*\* initialize \*/

myVR.begin(9600);

Serial.begin(115200);

Serial.println("Elechouse Voice Recognition V3 Module\r\nMulti Commands sample");

myservo1.attach(5); // attaches the servo on pin 5 to the servo object

myservo2.attach(10); // attaches the servo on pin 6 to the servo object

myservo3.attach(4); // attaches the servo on pin 4 to the servo object

pinMode(dcm1, OUTPUT);

pinMode(dcm2, OUTPUT);

pinMode(led1, OUTPUT);

pinMode(led2, OUTPUT);

pinMode(led3, OUTPUT);

pinMode(buzz1, OUTPUT);

pinMode(buzz2, OUTPUT);

if(myVR.clear() == 0){

Serial.println("Recognizer cleared.");

}else{

Serial.println("Not find VoiceRecognitionModule.");

Serial.println("Please check connection and restart Arduino.");

while(1);

}

record[0] = oneRecord;

record[1] = twoRecord;

record[2] = threeRecord;

record[3] = fourRecord;

record[4] = fiveRecord;

record[5] = sixRecord;

record[6] = sevenRecord;

group = 0;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}

void loop()

{

int ret;

ret = myVR.recognize(buf, 50);

if(ret>0){

switch(buf[1]){

case oneRecord:

/\*\* turn on LED \*/

if(digitalRead(led) == HIGH){

digitalWrite(led, LOW);

}else{

digitalWrite(led, HIGH);

}

if(group == 0){

group = 1;

myVR.clear();

record[0] = oneRecord;

record[1] = group1Record1;

record[2] = group1Record2;

record[3] = group1Record3;

record[4] = group1Record4;

record[5] = group1Record5;

record[6] = group1Record6;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}else{

group = 0;

myVR.clear();

record[0] = oneRecord;

record[1] = twoRecord;

record[2] = threeRecord;

record[3] = fourRecord;

record[4] = fiveRecord;

record[5] = sixRecord;

record[6] = sevenRecord;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}

break;

case twoRecord:

/\*\* turn on LED \*/

if(digitalRead(led) == HIGH){

digitalWrite(led, LOW);

}else{

digitalWrite(led, HIGH);

}

if(group == 0){

group = 2;

myVR.clear();

record[0] = twoRecord;

record[1] = group2Record1;

record[2] = group2Record2;

record[3] = group2Record3;

record[4] = group2Record4;

record[5] = group2Record5;

record[6] = group2Record6;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}else{

group = 0;

myVR.clear();

record[0] = oneRecord;

record[1] = twoRecord;

record[2] = threeRecord;

record[3] = fourRecord;

record[4] = fiveRecord;

record[5] = sixRecord;

record[6] = sevenRecord;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}

break;

case threeRecord:

/\*\* turn on LED \*/

if(digitalRead(led) == HIGH){

digitalWrite(led, LOW);

}else{

digitalWrite(led, HIGH);

}

if(group == 0){

group = 3;

myVR.clear();

record[0] = threeRecord;

record[1] = group3Record1;

record[2] = group3Record2;

record[3] = group3Record3;

record[4] = group3Record4;

record[5] = group3Record5;

record[6] = group3Record6;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}else{

group = 0;

myVR.clear();

record[0] = oneRecord;

record[1] = twoRecord;

record[2] = threeRecord;

record[3] = fourRecord;

record[4] = fiveRecord;

record[5] = sixRecord;

record[6] = sevenRecord;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}

break;

case fourRecord:

/\*\* turn on LED \*/

if(digitalRead(led) == HIGH){

digitalWrite(led, LOW);

}else{

digitalWrite(led, HIGH);

}

if(group == 0){

group = 4;

myVR.clear();

record[0] = fourRecord;

record[1] = group4Record1;

record[2] = group4Record2;

record[3] = group4Record3;

record[4] = group4Record4;

record[5] = group4Record5;

record[6] = group4Record6;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}else{

group = 0;

myVR.clear();

record[0] = oneRecord;

record[1] = twoRecord;

record[2] = threeRecord;

record[3] = fourRecord;

record[4] = fiveRecord;

record[5] = sixRecord;

record[6] = sevenRecord;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}

break;

case fiveRecord:

/\*\* turn on LED \*/

if(digitalRead(led) == HIGH){

digitalWrite(led, LOW);

}else{

digitalWrite(led, HIGH);

}

if(group == 0){

group = 5;

myVR.clear();

record[0] = fiveRecord;

record[1] = group5Record1;

record[2] = group5Record2;

record[3] = group5Record3;

record[4] = group5Record4;

record[5] = group5Record5;

record[6] = group5Record6;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}else{

group = 0;

myVR.clear();

record[0] = oneRecord;

record[1] = twoRecord;

record[2] = threeRecord;

record[3] = fourRecord;

record[4] = fiveRecord;

record[5] = sixRecord;

record[6] = sevenRecord;

if(myVR.load(record, 7) >= 0){

printRecord(record, 7);

Serial.println(F("loaded."));

}

}

break;

case group1Record1:

/\*\* turn on servo \*/

for(pos = 0; pos <= 90 ; pos += 1) // goes from 0 degrees to 180 degrees

{ // in steps of 1 degree

myservo1.write(pos); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

}

break;

case group1Record2:

/\*\* turn off servo\*/

for(pos = 90; pos>=0; pos-=1) // goes from 180 degrees to 0 degrees

{

myservo1.write(pos); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

}

break;

case group1Record3:

/\*\* turn on buzzer 1 - guest \*/

digitalWrite(buzz1, HIGH);

break;

case group1Record4:

/\*\* turn on buzzer 1 - guest \*/

digitalWrite(buzz1, LOW);

break;

case group2Record1:

/\*\* turn on led 1 - kitchen \*/

digitalWrite(led1, HIGH);

break;

case group2Record2:

/\*\* turn off led 1 - kitchen\*/

digitalWrite(led1, LOW);

break;

case group2Record3:

/\*\*brighten up led 2 - hall\*\*/

analogWrite(led2, brightness); // change the brightness for next time through the loop:

Serial.println(brightness);

brightness = brightness + fadeAmount;

if (brightness == 0 || brightness == 250) {

fadeAmount = -fadeAmount ;

}

break;

case group2Record4:

/\*\* turn off led 2 - hall\*/

analogWrite(led2, 0);

break;

case group2Record5:

/\*\* turn on led 3 - room\*/

digitalWrite(led3, HIGH);

break;

case group2Record6:

/\*\* turn off led 3 - room\*/

digitalWrite(led3, LOW);

break;

case group3Record1:

/\*\* turn on fan\*/

digitalWrite(dcm1, HIGH);

break;

case group3Record2:

/\*\* turn off fan\*/

digitalWrite(dcm1, LOW);

break;

case group4Record1:

/\*\* turn on servo- curtain\*/

for(pos2 = 0; pos2 <= 180 ; pos2 += 1) // goes from 0 degrees to 180 degrees

{ // in steps of 1 degree

myservo3.write(pos2); // tell servo to go to position in variable 'pos'

delay(5);

}

for(pos1 = 0; pos1 <= 180 ; pos1 += 1) // goes from 0 degrees to 180 degrees

{ // in steps of 1 degree

myservo2.write(pos1); // tell servo to go to position in variable 'pos'

delay(5);

}

break;

case group4Record2:

/\*\* turn off servo - curtain\*/

for(pos1 = 180; pos1>=0; pos1-=1) // goes from 180 degrees to 0 degrees

{

myservo2.write(pos1); // tell servo to go to position in variable 'pos'

delay(5);

}

for(pos2 = 180; pos2>=0; pos2-=1) // goes from 180 degrees to 0 degrees

{

myservo3.write(pos2); // tell servo to go to position in variable 'pos'

delay(5);

}

break;

case group4Record3:

/\*\* turn on sprinkler \*/

digitalWrite(dcm2, HIGH);

break;

case group4Record4:

/\*\* turn off sprinkler\*/

digitalWrite(dcm2, LOW);

break;

case group5Record1:

/\*\* turn on help siren\*/

digitalWrite(buzz2, HIGH);

break;

case group5Record2:

/\*\* turn off help siren\*/

digitalWrite(buzz2, LOW);

break;

case group5Record3:

/\*\* turn off alaram\*/

digitalWrite(led, LOW);

break;

default:

Serial.println("Record function undefined");

break;

}

/\*\* voice recognized \*/

printVR(buf);

}

}

/\*\*

@brief Print signature, if the character is invisible,

print hexible value instead.

@param buf --> command length

len --> number of parameters

\*/

void printSignature(uint8\_t \*buf, int len)

{

int i;

for(i=0; i<len; i++){

if(buf[i]>0x19 && buf[i]<0x7F){

Serial.write(buf[i]);

}

else{

Serial.print("[");

Serial.print(buf[i], HEX);

Serial.print("]");

}

}

}

/\*\*

@brief Print signature, if the character is invisible,

print hexible value instead.

@param buf --> VR module return value when voice is recognized.

buf[0] --> Group mode(FF: None Group, 0x8n: User, 0x0n:System

buf[1] --> number of record which is recognized.

buf[2] --> Recognizer index(position) value of the recognized record.

buf[3] --> Signature length

buf[4]~buf[n] --> Signature

\*/

void printVR(uint8\_t \*buf)

{

Serial.println("VR Index\tGroup\tRecordNum\tSignature");

Serial.print(buf[2], DEC);

Serial.print("\t\t");

if(buf[0] == 0xFF){

Serial.print("NONE");

}

else if(buf[0]&0x80){

Serial.print("UG ");

Serial.print(buf[0]&(~0x80), DEC);

}

else{

Serial.print("SG ");

Serial.print(buf[0], DEC);

}

Serial.print("\t");

Serial.print(buf[1], DEC);

Serial.print("\t\t");

if(buf[3]>0){

printSignature(buf+4, buf[3]);

}

else{

Serial.print("NONE");

}

// Serial.println("\r\n");

Serial.println();

}

void printRecord(uint8\_t \*buf, uint8\_t len)

{

Serial.print(F("Record: "));

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

Serial.print(buf[i], DEC);

Serial.print(", ");

}

}