#if defined(ARDUINO) && ARDUINO >= 100

#include "Arduino.h"

#include "SoftwareSerial.h"

SoftwareSerial port(12,13);

#else // Arduino 0022 - use modified NewSoftSerial

#include "WProgram.h"

#include "NewSoftSerial.h"

NewSoftSerial port(12,13);

#endif

#include "EasyVR.h"

EasyVR easyvr(port);

#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins

LiquidCrystal lcd(10, 11, 6, 7, 8, 9);

//Groups and Commands

enum Groups

{

GROUP\_0 = 0,

GROUP\_1 = 1,

};

enum Group0

{

G0\_ARDUINO = 0,

};

enum Group1

{

G1\_ONE\_ON = 0,

G1\_ONE\_OFF = 1,

G1\_TWO\_ON = 2,

G1\_TWO\_OFF = 3,

G1\_THREE\_ON = 4,

G1\_THREE\_OFF = 5,

G1\_FOUR\_ON = 6,

G1\_FOUR\_OFF = 7,

G1\_ALL\_ON = 8,

G1\_OFF\_ALL = 9,

};

EasyVRBridge bridge;

int8\_t group, idx;

void setup()

{

pinMode(2,OUTPUT);

pinMode(3,OUTPUT);

pinMode(4,OUTPUT);

pinMode(5,OUTPUT);

digitalWrite(2,LOW);

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,LOW);

lcd.begin(20,4);

// bridge mode?

if (bridge.check())

{

cli();

bridge.loop(0, 1, 12, 13);

}

// run normally

Serial.begin(9600);

port.begin(9600);

if (!easyvr.detect())

{

Serial.println("EasyVR not detected!");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("EasyVR not detected!");

for (;;);

}

easyvr.setPinOutput(EasyVR::IO1, LOW);

Serial.println("EasyVR detected!");

lcd.clear();

lcd.setCursor(0,0);

lcd.print("EasyVR detected!");

easyvr.setTimeout(5);

easyvr.setLanguage(0);

group = EasyVR::TRIGGER; //<-- start group (customize)

}

void action();

void loop()

{

easyvr.setPinOutput(EasyVR::IO1, HIGH); // LED on (listening)

Serial.print("Say a command in Group ");

lcd.setCursor(0,1);

lcd.print("Say a comm. in gp:");

Serial.println(group);

lcd.print(group);

easyvr.recognizeCommand(group);

do

{

// can do some processing while waiting for a spoken command

}

while (!easyvr.hasFinished());

easyvr.setPinOutput(EasyVR::IO1, LOW); // LED off

idx = easyvr.getWord();

if (idx >= 0)

{

// built-in trigger (ROBOT)

// group = GROUP\_X; <-- jump to another group X

return;

}

idx = easyvr.getCommand();

if (idx >= 0)

{

// print debug message

uint8\_t train = 0;

char name[32];

Serial.print("Command: ");

lcd.setCursor(0,2);

lcd.print("Command: ");

Serial.print(idx);

lcd.print(idx);

if (easyvr.dumpCommand(group, idx, name, train))

{

Serial.print(" = ");

lcd.print("=");

Serial.println(name);

lcd.print(name);

delay(700);

lcd.setCursor(0,2);

lcd.print(" ");

}

else

Serial.println();

easyvr.playSound(0, EasyVR::VOL\_FULL);

// perform some action

action();

}

else // errors or timeout

{

if (easyvr.isTimeout())

Serial.println("Timed out, try again...");

lcd.setCursor(0,2);

lcd.print("Time Out..");

delay(700);

lcd.setCursor(0,2);

lcd.print(" ");

int16\_t err = easyvr.getError();

if (err >= 0)

{

Serial.print("Error ");

lcd.print("Error ");

Serial.println(err, HEX);

lcd.print(err);

lcd.setCursor(0,2);

lcd.print(" ");

}

}

}

void action()

{

switch (group)

{

case GROUP\_0:

switch (idx)

{

case G0\_ARDUINO:

group=GROUP\_1;

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

}

break;

case GROUP\_1:

switch (idx)

{

case G1\_ONE\_ON:

Serial.println("device 1 on");

digitalWrite(2,HIGH);

lcd.setCursor(0,3);

lcd.print("device 1 on");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

case G1\_ONE\_OFF:

Serial.println("device 1 off");

digitalWrite(2,LOW);

lcd.setCursor(0,3);

lcd.print("device 1 off");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

case G1\_TWO\_ON:

Serial.println("device 2 on");

digitalWrite(3,HIGH);

lcd.setCursor(0,3);

lcd.print("device 2 on");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

case G1\_TWO\_OFF:

Serial.println("device 2 off");

digitalWrite(3,LOW);

lcd.setCursor(0,3);

lcd.print("device 2 off");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

case G1\_THREE\_ON:

Serial.println("device 3 on");

digitalWrite(4,HIGH);

lcd.setCursor(0,3);

lcd.print("device 3 on");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

case G1\_THREE\_OFF:

Serial.println("device 3 off");

digitalWrite(4,LOW);

lcd.setCursor(0,3);

lcd.print("device 3 off");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

case G1\_FOUR\_ON:

Serial.println("device 4 on");

digitalWrite(5,HIGH);

lcd.setCursor(0,3);

lcd.print("device 4 on");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

case G1\_FOUR\_OFF:

Serial.println("device 4 off");

digitalWrite(5,LOW);

lcd.setCursor(0,3);

lcd.print("device 4 off");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

case G1\_ALL\_ON:

Serial.println("all devices on");

digitalWrite(2,HIGH);

digitalWrite(3,HIGH);

digitalWrite(4,HIGH);

digitalWrite(5,HIGH);

lcd.setCursor(0,3);

lcd.print("All devices on");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

case G1\_OFF\_ALL:

Serial.println("all devices off");

digitalWrite(2,LOW);

digitalWrite(3,LOW);

digitalWrite(4,LOW);

digitalWrite(5,LOW);

lcd.setCursor(0,3);

lcd.print("All devices off");

delay(700);

lcd.setCursor(0,3);

lcd.print(" ");

// write your action code here

// group = GROUP\_X; <-- or jump to another group X for composite commands

break;

}

break;

}

}