به نام خدا

امیررضا رجبی 9831126

پروژه آز و آز اسمبلی 1

پروژه آز :

برای انتقال آبشاری از دی فلیپ فلاپ استفاده کردم

برای هر چهار ردیف یک برد اسفاده کردم یعنی در مجموع از 4 (اسلیو) + 1(مستر) استفاده کردم نحوه ساختن حروف هم از عکس "حروف" استفاده کردم

Master’s coed:

#include <SPI.h>

#define SS0 47

#define SS1 46

#define SS2 45

#define SS3 44

int PINs[4] = {SS0, SS1, SS2, SS3};

char M[100];

int index = 0;

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

for(int i = 0; i < 4; i++){

pinMode(PINs[i], OUTPUT);

digitalWrite(PINs[i], HIGH);

}

SPI.begin();

}

void sendData(int i){

digitalWrite(PINs[i], LOW);

SPI.transfer(M[i]);

digitalWrite(PINs[i], HIGH);

}

void loop() {

// put your main code here, to run repeatedly:

if (Serial.available() > 0) {

char c = Serial.read();

if (c == '-') {

index = 0;

for (int i = 0; i < 4; i++)

sendData(i);

}

else {

M[index] = c;

index++;

}

}

}

Slave’s code:

#include <SPI.h>

#define LED0 7

#define LED1 6

#define LED2 5

#define LED3 4

#define MISO 50

#define MOSI 51

#define SCK 52

#define SS 53

#define t 50

int LEDs[] = {LED0, LED1, LED2, LED3};

char M[10];

volatile boolean isFinished = false;

void setup() {

// put your setup code here, to run once:

for (int i = 0; i < 4; i++) {

pinMode(LEDs[i], OUTPUT);

}

pinMode(MISO, OUTPUT);

pinMode(MOSI, INPUT);

pinMode(SCK, INPUT);

pinMode(SS, INPUT\_PULLUP);

SPCR |= \_BV(SPE);

SPI.attachInterrupt();

}

void displayLetter(int i0, int i1, int i2, int i3, int delayTime) {

digitalWrite(LED0, i0);

digitalWrite(LED1, i1);

digitalWrite(LED2, i2);

digitalWrite(LED3, i3);

delay(delayTime);

}

void displayAlphabets(char c){

if (c == 'A'){

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'B'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

}

else if (c == 'C'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'D'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

}

else if (c == 'E'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'F'){

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'G'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'H'){

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

}

else if (c == 'I'){

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

}

else if (c == 'J'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

}

else if (c == 'K'){

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'L'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

}

else if (c == 'M'){

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'N'){

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'O'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'P'){

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'Q'){

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'R'){

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'S'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

else if (c == 'T'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, LOW, t);

}

else if (c == 'U'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, LOW, t);

}

else if (c == 'V'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

}

else if (c == 'W'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

}

else if (c == 'X'){

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

}

else if (c == 'Y'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

displayLetter(HIGH, LOW, LOW, HIGH, t);

}

else if (c == 'Z'){

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(LOW, LOW, LOW, LOW, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(LOW, LOW, LOW, HIGH, t);

displayLetter(HIGH, HIGH, HIGH, HIGH, t);

}

displayLetter(LOW, LOW, LOW, LOW, t);

}

void loop() {

// put your main code here, to run repeatedly:

if (isFinished){

displayAlphabets(M[0]);

isFinished = false;

}

}

ISR (SPI\_STC\_vect)

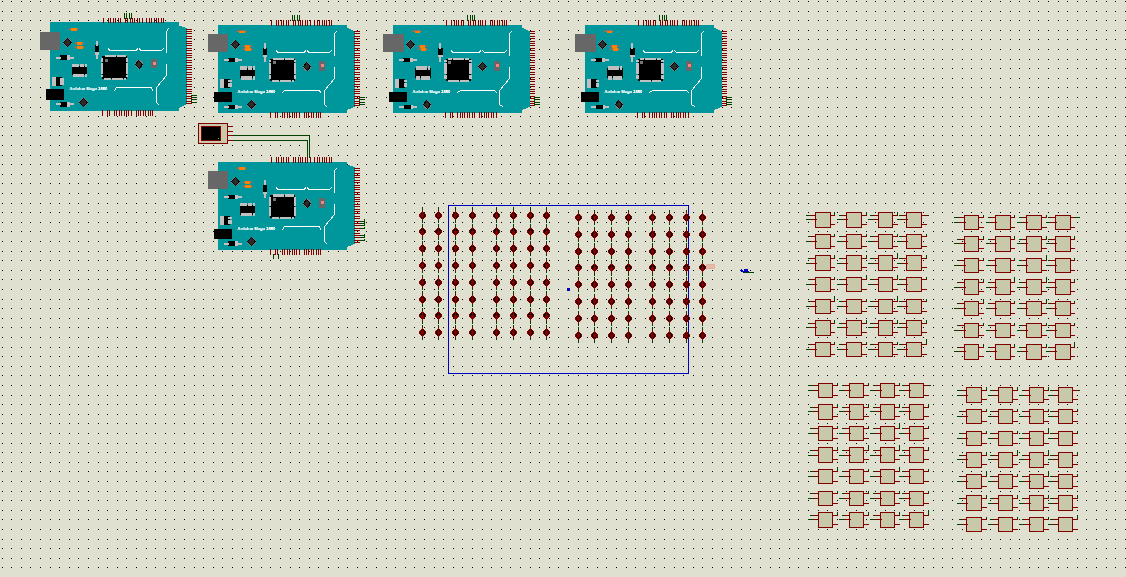
{

byte c = SPDR;

M[0] = c;

isFinished = true;

}



آز اسمبلی 1:

