1/23/2022

**به نام خدا**

گزارش پروژه نهایی درس ریز پردازنده پروژه شماره یک: XO

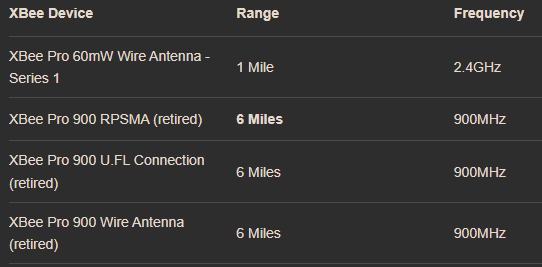
محمد جواد زندیه 9831032

دانشگاه صنعتی امیرکبیر دانشکده مهندسی کامپیوتر

در این پروژه قرار است تا به وسیله ماژول Xbee و پروتکول استاندارد ZigBee یک بازی XO را پیاده سازی کنیم که دو کاربر بتوانند با فاصله ای حداکثر 6 مایل از هم بازی کنند.

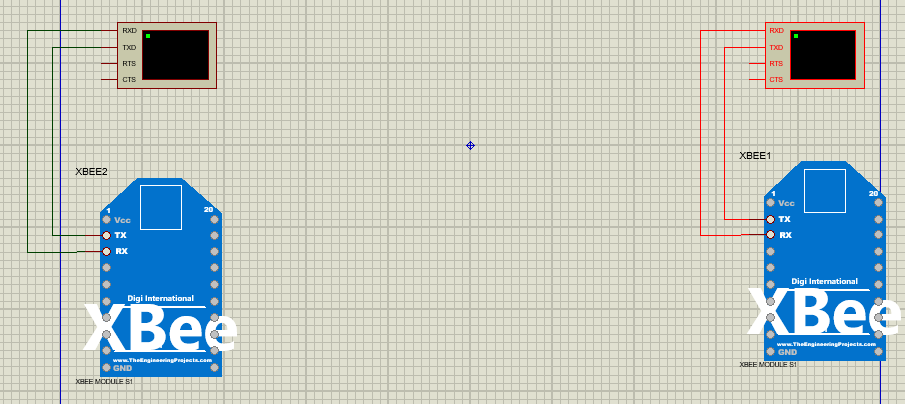
محدوده کاری (range) ماژول ایکس بی معمولا چقدر است؟

بسته به دیوایس های مختلف این محدوده متفاوت است اما معمولا حدود 6 مایل می باشد و با فرکانس 900 MHz.



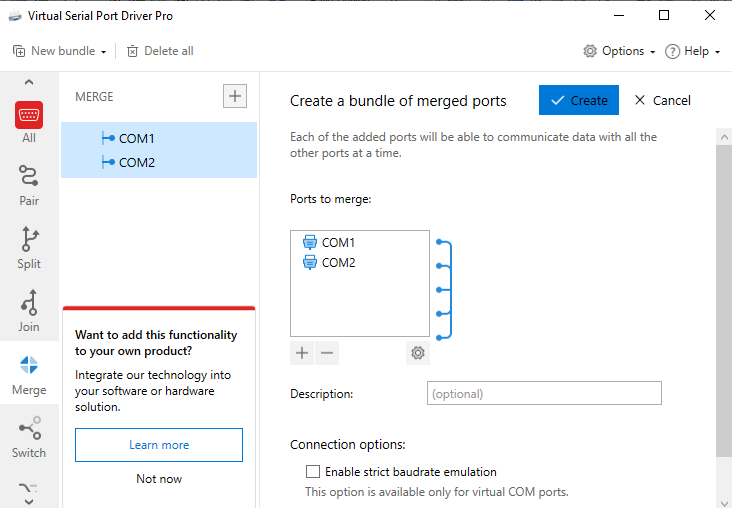
گام اول:

دو ماژول Xbee را هر کدام به یک ترمینال مجازی متصل می کنیم

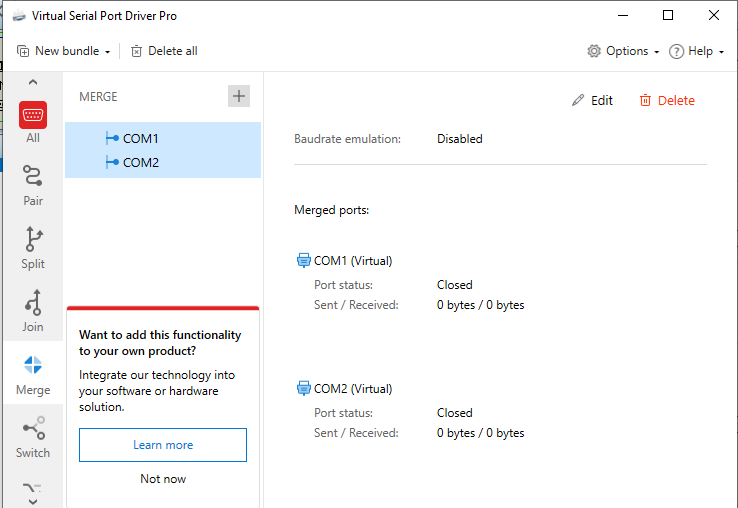


در تنظیمات ماژول ها، یکی را به عنوان Com1 و دیگری را به عنوان Com2 قرار میدهیم. همچنین virtual baud rate ره دو را روی 9600 قرار می دهیم.

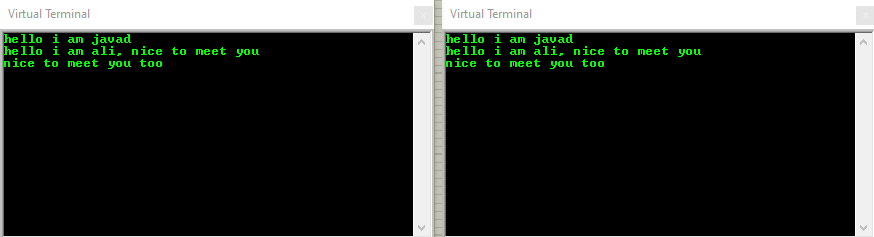
با استفاده از نرم افزار [Virtual Serial Port Driver](https://www.eltima.com/products/vspdxp/?gclid=CjwKCAiAlrSPBhBaEiwAuLSDUGnasHtxahGKdTjDhs38sYhPARHsokBaemWJSuRfi9SujXyTotHYYRoCWLsQAvD_BwE) دو پورت Com1 و Com2 را در قسمت merge تلفیق می کنیم.

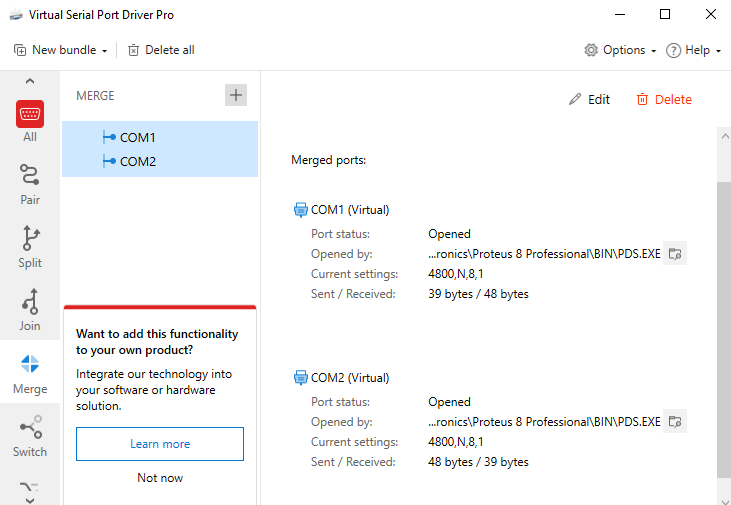


پس از merge کردن به این وضعیت می رسیم:

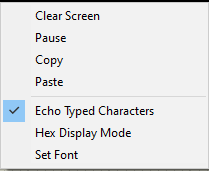


حال به تست انتقال می پردازیم:



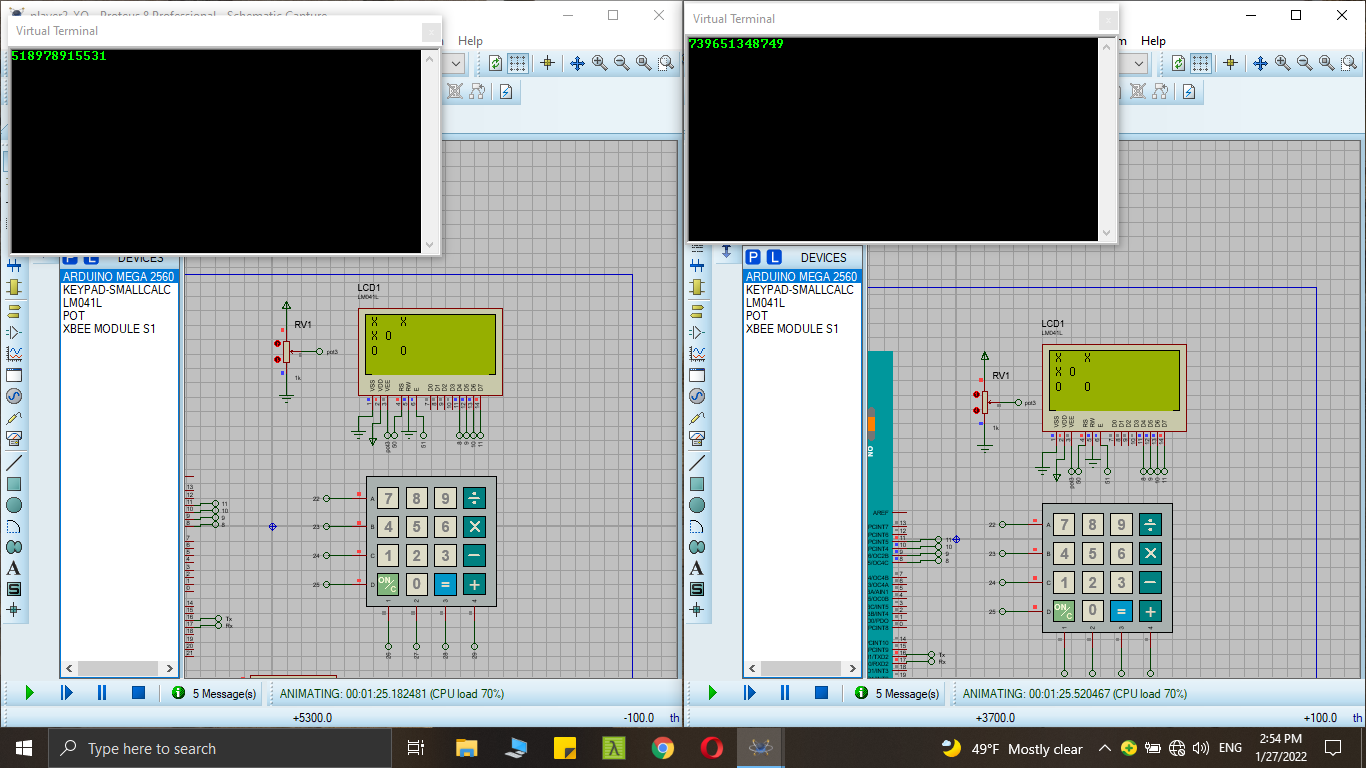


همان طور که مشاهده می شود، خط اول و سوم توسط ترمینال سمت چپ ارسال شده است و خط دوم هم توسط ترمینال سمت راست. توجه شود که برای آنکه پیام برای هر دو قابل مشاهده باشد باید با راست کلیک روی هر دو ترمینال گزینه Echo را فعال کنید. اگر این گزینه فعال نباشد، پیغام فقط در گیرنده قابل مشاهده است و خود فرستنده تغییری در ترمینال اش مشاهده نمی شود.

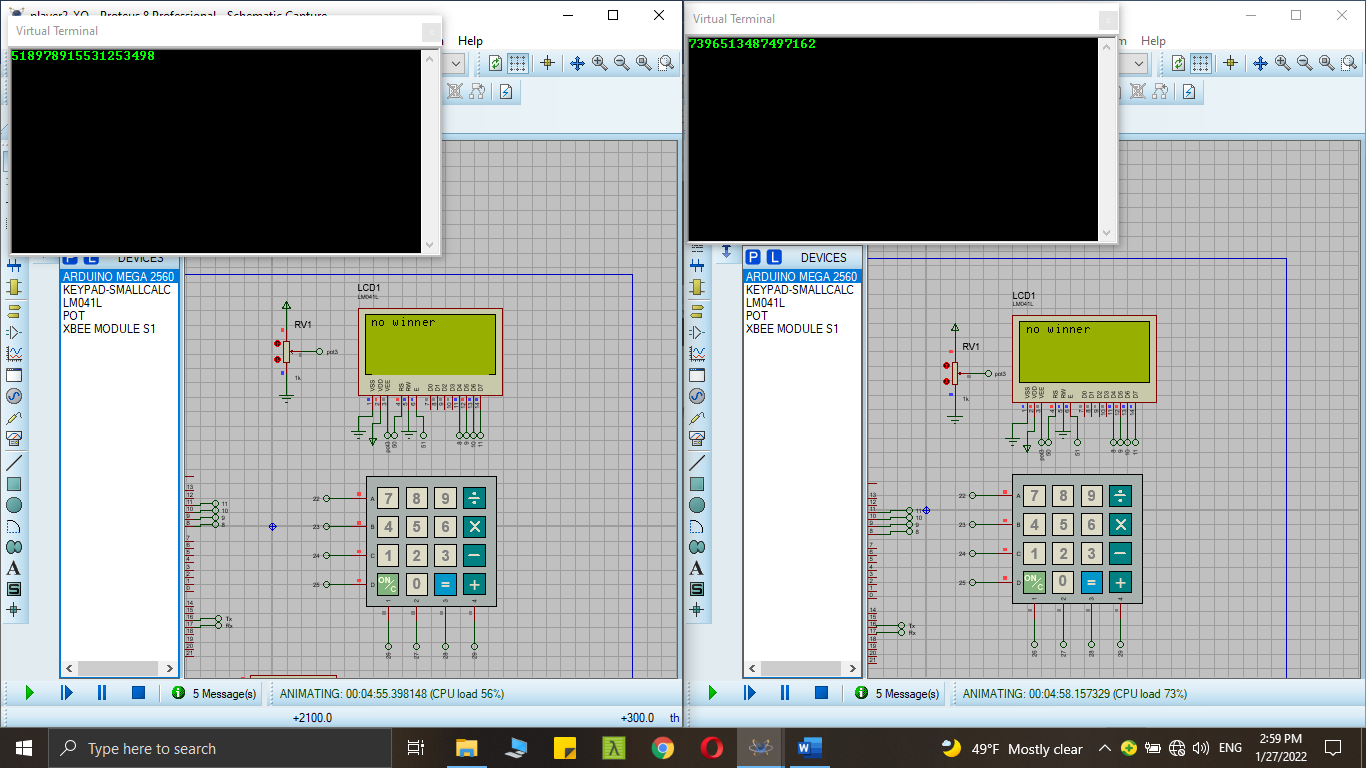


گام دوم:

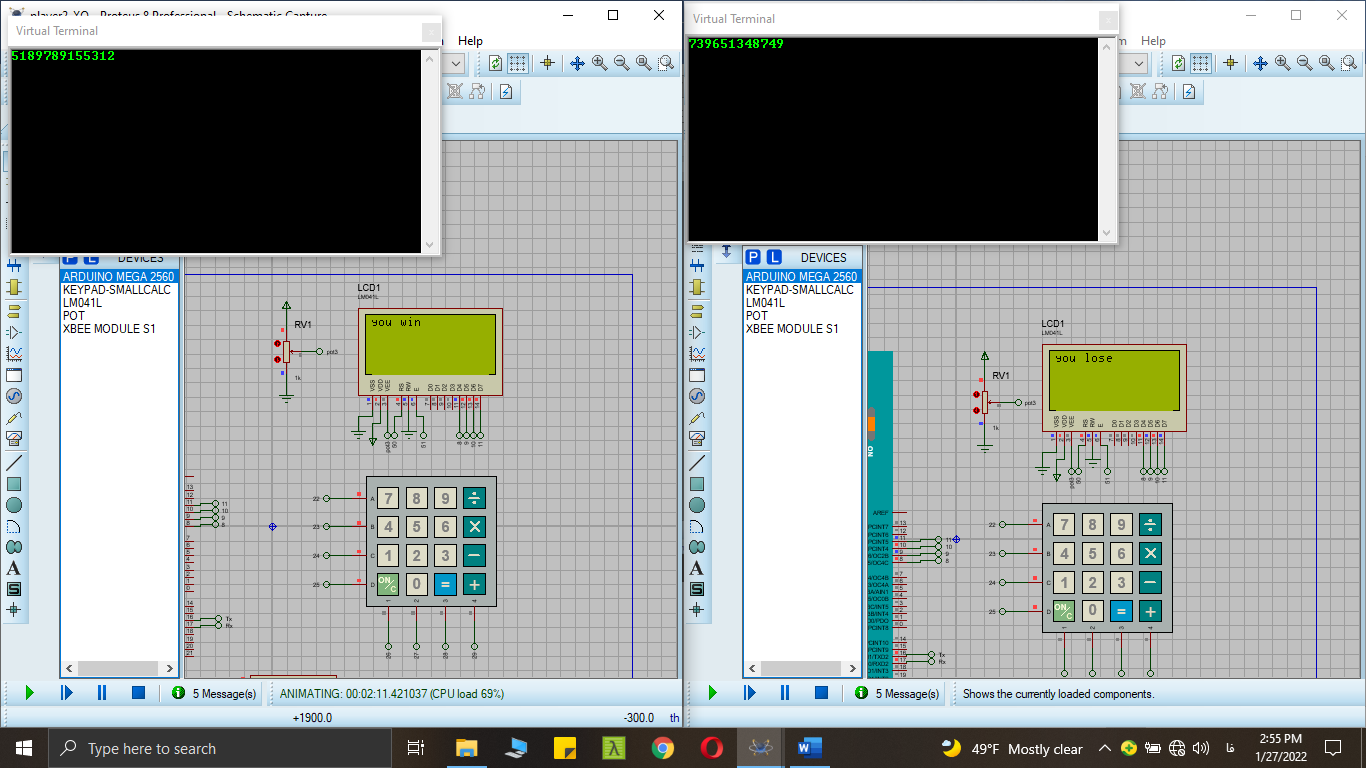
عکس از صفحه بازی



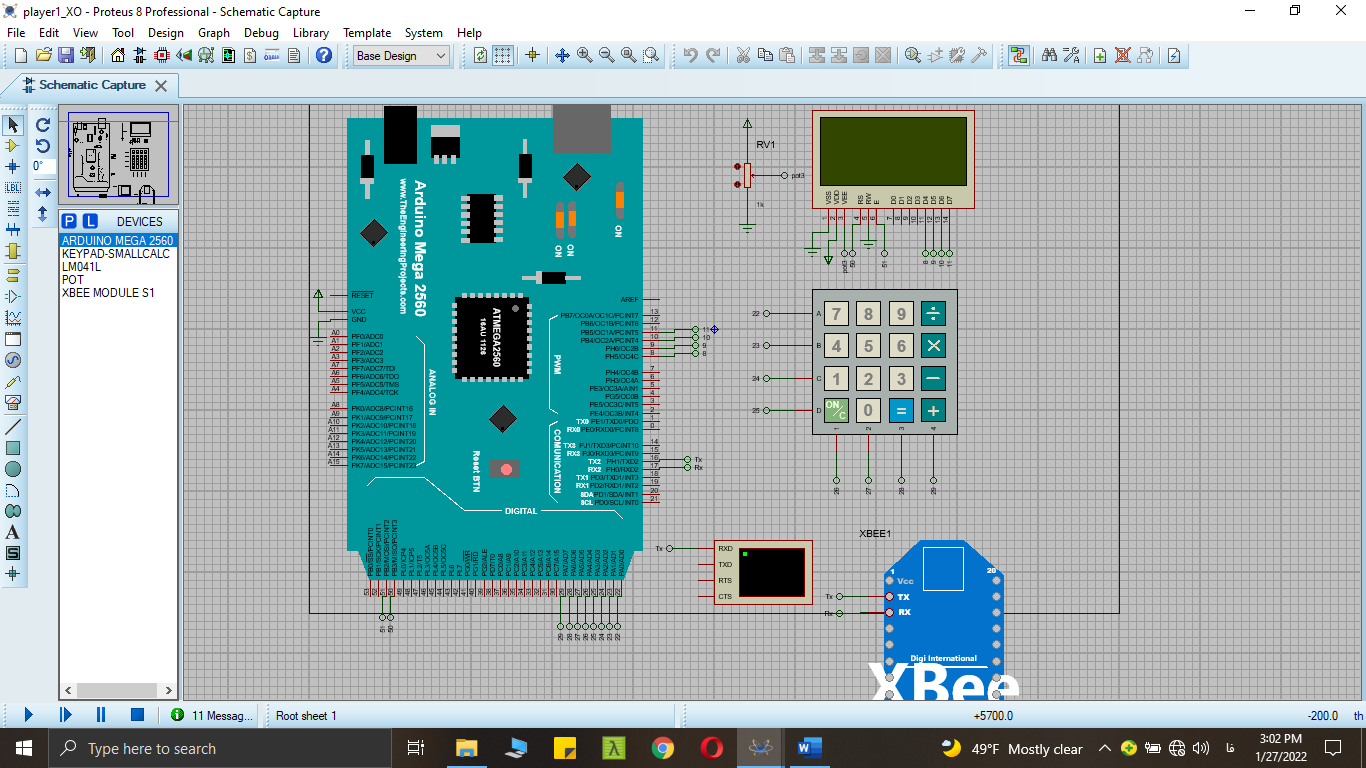
در حالت تساوی : no winner چاپ شده و بعد از دو ثانیه بازی از اول سر گرفته می شود و نوبت هم با بازیکن 1 است.



در حالت برد: هر یک از بازیکنان که برنده شوند، برایشان وضعیت شان نمایش داده خواهد شد و پس از دو ثانیه دوباره صفحه ریست شده و نوبت هم به کسی می رسد که در بازی قبل برنده شده است.



عکس از مدار:



کد Arduino برای player1 :

// player 1

#include <LiquidCrystal.h>

#include <Keypad.h>

// lcd object initializing

#define RS\_PIN 50

#define EN\_PIN 51

#define D4\_PIN 8

#define D5\_PIN 9

#define D6\_PIN 10

#define D7\_PIN 11

LiquidCrystal lcd(RS\_PIN, EN\_PIN, D4\_PIN, D5\_PIN, D6\_PIN, D7\_PIN);

// keypad object initializing

const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

{'7','8','9','/'},

{'4','5','6','\*'},

{'1','2','3','-'},

{'C','0','=','+'}

};

byte rowPins[ROWS] = {22, 23, 24, 25};

byte colPins[COLS] = {26, 27, 28, 29};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// point

char player1\_point = 'X';

char player2\_point = 'O';

// grid fill places

int fill[10];

// 1: player1 turn

// 2: player2 turn

int turn = 1;

void setup() {

Serial2.begin(9600);

keypad.addEventListener(keypadEvent);

lcd.begin(16, 4);

lcd.clear();

for(int i=0; i<10; i++)

fill[i] = 0;

}

void loop() {

// keypad pressed button

char c = keypad.getKey();

if(Serial2.available()){

char data = Serial2.read();

int num = data - '0';

update\_screen(num, 2); // show what player 2 done

}

}

void keypadEvent(KeypadEvent key){

if(keypad.getState() == PRESSED){

if((key >= '1') & (key <= '9')){

int num = key - '0';

if((fill[num] == 0) & (turn == 1)){

Serial2.print(key);

update\_screen(num, 1); // show what player 1 done

}

}

}

}

void update\_screen(int num, int player){

switch(num){

case 1:

lcd.setCursor(0, 2);

break;

case 2:

lcd.setCursor(2, 2);

break;

case 3:

lcd.setCursor(4, 2);

break;

case 4:

lcd.setCursor(0, 1);

break;

case 5:

lcd.setCursor(2, 1);

break;

case 6:

lcd.setCursor(4, 1);

break;

case 7:

lcd.setCursor(0, 0);

break;

case 8:

lcd.setCursor(2, 0);

break;

case 9:

lcd.setCursor(4, 0);

break;

}

if(player == 1){

fill[num] = 1;

turn = 2;

lcd.print(player1\_point);

}

else if(player == 2){

fill[num] = 2;

turn = 1;

lcd.print(player2\_point);

}

check\_win();

}

void check\_win(){

int winner = 0;

if( (fill[1]\*fill[2]\*fill[3] == 1) ||

(fill[4]\*fill[5]\*fill[6] == 1) ||

(fill[7]\*fill[8]\*fill[9] == 1) ||

(fill[1]\*fill[4]\*fill[7] == 1) ||

(fill[2]\*fill[5]\*fill[8] == 1) ||

(fill[3]\*fill[6]\*fill[9] == 1) ||

(fill[1]\*fill[5]\*fill[9] == 1) ||

(fill[3]\*fill[5]\*fill[7] == 1)){

winner = 1;

}

else if( (fill[1]\*fill[2]\*fill[3] == 8) ||

(fill[4]\*fill[5]\*fill[6] == 8) ||

(fill[7]\*fill[8]\*fill[9] == 8) ||

(fill[1]\*fill[4]\*fill[7] == 8) ||

(fill[2]\*fill[5]\*fill[8] == 8) ||

(fill[3]\*fill[6]\*fill[9] == 8) ||

(fill[1]\*fill[5]\*fill[9] == 8) ||

(fill[3]\*fill[5]\*fill[7] == 8)){

winner = 2;

}

if(winner != 0){

delay(10);

lcd.clear();

if(winner == 1)

lcd.print("you win");

else

lcd.print("you lose");

delay(2000);

reset\_game(winner);

}else{

int mul = 1;

for(int i=1; i<10; i++)

mul \*= fill[i];

if(mul != 0){

lcd.clear();

lcd.print("no winner");

delay(2000);

reset\_game(1);

}

}

}

void reset\_game(int winner){

lcd.clear();

for(int i=0; i<10; i++)

fill[i] = 0;

turn = winner;

}

کد Arduino برای player2 :

// player 2

#include <LiquidCrystal.h>

#include <Keypad.h>

// lcd object initializing

#define RS\_PIN 50

#define EN\_PIN 51

#define D4\_PIN 8

#define D5\_PIN 9

#define D6\_PIN 10

#define D7\_PIN 11

LiquidCrystal lcd(RS\_PIN, EN\_PIN, D4\_PIN, D5\_PIN, D6\_PIN, D7\_PIN);

// keypad object initializing

const byte ROWS = 4;

const byte COLS = 4;

char keys[ROWS][COLS] = {

{'7','8','9','/'},

{'4','5','6','\*'},

{'1','2','3','-'},

{'C','0','=','+'}

};

byte rowPins[ROWS] = {22, 23, 24, 25};

byte colPins[COLS] = {26, 27, 28, 29};

Keypad keypad = Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS);

// point

char player1\_point = 'X';

char player2\_point = 'O';

// grid fill places

int fill[10];

// 1: player1 turn

// 2: player2 turn

int turn = 1;

void setup() {

Serial2.begin(9600);

keypad.addEventListener(keypadEvent);

lcd.begin(16, 4);

lcd.clear();

for(int i=0; i<10; i++)

fill[i] = 0;

}

void loop() {

// keypad pressed button

char c = keypad.getKey();

if(Serial2.available()){

char data = Serial2.read();

int num = data - '0';

update\_screen(num, 1); // show what player 1 done

}

}

void keypadEvent(KeypadEvent key){

if(keypad.getState() == PRESSED){

if((key >= '1') & (key <= '9')){

int num = key - '0';

if((fill[num] == 0) & (turn == 2)){

Serial2.print(key);

update\_screen(num, 2); // show what player 2 done

}

}

}

}

void update\_screen(int num, int player){

switch(num){

case 1:

lcd.setCursor(0, 2);

break;

case 2:

lcd.setCursor(2, 2);

break;

case 3:

lcd.setCursor(4, 2);

break;

case 4:

lcd.setCursor(0, 1);

break;

case 5:

lcd.setCursor(2, 1);

break;

case 6:

lcd.setCursor(4, 1);

break;

case 7:

lcd.setCursor(0, 0);

break;

case 8:

lcd.setCursor(2, 0);

break;

case 9:

lcd.setCursor(4, 0);

break;

}

if(player == 1){

fill[num] = 1;

turn = 2;

lcd.print(player1\_point);

}

else if(player == 2){

fill[num] = 2;

turn = 1;

lcd.print(player2\_point);

}

check\_win();

}

void check\_win(){

int winner = 0;

if( (fill[1]\*fill[2]\*fill[3] == 1) ||

(fill[4]\*fill[5]\*fill[6] == 1) ||

(fill[7]\*fill[8]\*fill[9] == 1) ||

(fill[1]\*fill[4]\*fill[7] == 1) ||

(fill[2]\*fill[5]\*fill[8] == 1) ||

(fill[3]\*fill[6]\*fill[9] == 1) ||

(fill[1]\*fill[5]\*fill[9] == 1) ||

(fill[3]\*fill[5]\*fill[7] == 1)){

winner = 1;

}

else if( (fill[1]\*fill[2]\*fill[3] == 8) ||

(fill[4]\*fill[5]\*fill[6] == 8) ||

(fill[7]\*fill[8]\*fill[9] == 8) ||

(fill[1]\*fill[4]\*fill[7] == 8) ||

(fill[2]\*fill[5]\*fill[8] == 8) ||

(fill[3]\*fill[6]\*fill[9] == 8) ||

(fill[1]\*fill[5]\*fill[9] == 8) ||

(fill[3]\*fill[5]\*fill[7] == 8)){

winner = 2;

}

if(winner != 0){

delay(10);

lcd.clear();

if(winner == 2)

lcd.print("you win");

else

lcd.print("you lose");

delay(2000);

reset\_game(winner);

}else{

int mul = 1;

for(int i=1; i<10; i++)

mul \*= fill[i];

if(mul != 0){

lcd.clear();

lcd.print("no winner");

delay(2000);

reset\_game(1);

}

}

}

void reset\_game(int winner){

lcd.clear();

for(int i=0; i<10; i++)

fill[i] = 0;

turn = winner;

}