#include<stdio.h>

#include<conio.h>

#include<dos.h>

#include<graphics.h>

#include<stdlib.h>

#include<time.h>

#include<math.h>

raspberry\_pi();

raspberry\_connect();

laser();

hsensor();

int main()

{

int gd=DETECT,gm;

clrscr();

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

printf("Press a key to turn on RaspberryPi");

getch();

clrscr();

cleardevice();

raspberry\_pi();

raspberry\_connect();

getch();

clrscr();

cleardevice();

return 0;

}

raspberry\_connect()

{

int option;

cleardevice();

printf("Select the function:\n1.Heartbeat Monitor\n2.LASER\n");

scanf("%d",&option);

printf("\nPress Enter to connect the wires\n");

switch(option)

{

case 1:

getch();

cleardevice();

setcolor(11); //light blue wire connected to breadboard

line(112,15,112,65);

line(113,15,113,65);

line(114,15,114,65);

setcolor(3); //aqua blue wire connected to breadboard

line(160,15,160,65);

line(161,15,161,65);

line(162,15,162,65);

setcolor(YELLOW); //yellow wire connected to breadboard

line(143,15,143,55);

line(144,15,144,55);

line(145,15,145,55);

setcolor(12); //orange wire connected to breadboard

line(256,15,256,65);

line(257,15,257,65);

line(258,15,258,65);

setcolor(BROWN); //brown wire connected to breadboard

line(272,15,272,65);

line(273,15,273,65);

line(274,15,274,65);

setcolor(WHITE); //white wire connected to breadboard

line(287,15,287,65);

line(288,15,288,65);

line(289,15,289,65);

setcolor(5); //pink wire connected to breadboard

line(336,15,336,65);

line(337,15,337,65);

line(338,15,338,65);

setcolor(RED);

line(430,3,639,3);

line(430,4,638,4);

line(430,5,637,5);

line(639,3,639,30);

line(638,4,638,30);

line(637,5,637,30);

setcolor(GREEN);

line(430,6,637,6);

line(430,7,636,7);

line(430,8,635,8);

line(637,6,637,30);

line(636,7,636,30);

line(635,8,635,30);

setcolor(BLUE);

line(430,9,635,9);

line(430,10,634,10);

line(430,11,633,11);

line(635,9,635,30);

line(634,10,634,30);

line(633,11,633,30);

setcolor(3);

line(620,400,639,400);

line(620,401,638,401);

line(620,402,637,402);

line(637,402,637,600);

line(638,401,638,600);

line(639,400,639,600);

setcolor(WHITE);

rectangle(625,30,639,100);

rectangle(90,0,440,15);

outtextxy(220,5,"BREADBOARD");

outtextxy(630,35,"S");

outtextxy(630,45,"E");

outtextxy(630,55,"N");

outtextxy(630,65,"S");

outtextxy(630,75,"O");

outtextxy(630,85,"R");

outtextxy(615,470,"PC");

raspberry\_pi();

cleardevice();

hsensor();

break;

case 2:

getch();

cleardevice();

setcolor(YELLOW); //yellow wire connected to breadboard

line(304,10,304,55);

line(305,11,305,55);

line(306,12,306,55);

line(306,12,631,12);

line(305,11,632,11);

line(304,10,633,10);

line(631,12,631,50);

line(632,11,632,50);

line(633,10,633,50);

setcolor(12); //orange wire connected to breadboard

line(352,13,352,65);

line(353,14,353,65);

line(354,15,354,65);

line(354,15,628,15);

line(353,14,629,14);

line(352,13,630,13);

line(630,13,630,50);

line(629,14,629,50);

line(628,15,628,50);

setcolor(WHITE);

rectangle(625,50,639,105);

outtextxy(630,55,"L");

outtextxy(630,65,"A");

outtextxy(630,75,"S");

outtextxy(630,85,"E");

outtextxy(630,95,"R");

raspberry\_pi();

cleardevice();

laser();

setcolor(YELLOW); //yellow wire connected to Raspberry Pi

line(2,148,25,148);

line(1,149,25,149);

line(0,150,25,150);

line(2,148,2,0);

line(1,149,1,0);

line(0,150,0,0);

setcolor(12); //orange wire connected to Raspberry Pi

line(3,147,25,147);

line(4,146,25,146);

line(5,145,25,145);

line(3,147,3,0);

line(4,146,4,0);

line(5,145,5,0);

setcolor(WHITE);

outtextxy(30,20,"Raspberry Pi");

break;

default:

printf("Wrong input\n");

}

}

hsensor()

{

int gvx1=0,gvy1=0,gvx2=0,gvy2=450; //graph verticle

int ghx1=0,ghy1=0,ghx2=750,ghy2=0; //graph horizontal

int i=0; //test conditions

int x1=0,y1=250, x2=rand()%(80+1-1)+1 , y2=rand()% (450+1-10)+10;

int a1=1,b1=250, a2=x2+1, b2=y2;

setcolor(GREEN);

line(x1,y1,x2,y2);

line(a1,b1,a2,b2);

setcolor(WHITE); //initial graph

while(i<15)

{

line(gvx1,gvy1,gvx2,gvy2);

line(ghx1,ghy1,ghx2,ghy2);

gvx1+=50;

gvx2=gvx1;

ghy1+=50;

ghy2=ghy1;

i++;

}

while(!kbhit())

{

setcolor(YELLOW);

delay(320);

x1=x2;

y1=y2;

x2=rand() % (30+1-1)+x1;

y2=rand() % (380+1-10)+30;

a1=x1+1;

a2=x2+1;

b1=y1;

b2=y2;

setcolor(GREEN);

line(x1,y1,x2,y2);

line(a1,b1,a2,b2);

setcolor(RED);

outtextxy(350,460,"Press any key to KILL!");

if(ghy1>=450)

{

ghy1=500;

ghy2=500;

continue;

}

}

setcolor(RED);

x1=x2;

y1=y2;

a1=x1+1;

a2=x2+1;

b1=y1;

b2=y2;

line(x1,y1,x1,250);

line(a1,b1,a1,251);

line(x1,250,x1+1000,250);

line(a1,251,a1+1000,251);

getch();

return 0; //e\_initial graph

}

raspberry\_pi()

{

rectangle(100,40,430,80);

setcolor(YELLOW);

outtextxy(110,50,". . . . . . . . . . . . . . . . . . . .");

outtextxy(110,60,". . . . . . . . . . . . . . . . . . . .");

setcolor(WHITE);

circle(70,60,20);

circle(70,60,8);

circle(460,60,20);

circle(460,60,8);

rectangle(80,100,95,106);

rectangle(20,90,25,105);

rectangle(120,90,135,96);

rectangle(120,100,135,106);

rectangle(120,110,135,116);

rectangle(220,90,235,96);

rectangle(20,115,25,130);

rectangle(50,145,70,165);

rectangle(80,110,95,116);

circle(103,158,5);

circle(60,155,5);

circle(30,155,10);

circle(30,155,4);

circle(80,200,10);

circle(80,200,4);

circle(30,420,20);

circle(30,420,8);

rectangle(20,180,40,340);

rectangle(40,185,50,190);

rectangle(40,195,50,200);

rectangle(40,205,50,210);

rectangle(40,215,50,220);

rectangle(40,225,50,230);

rectangle(40,235,50,240);

rectangle(40,245,50,250);

rectangle(40,255,50,260);

rectangle(40,265,50,270);

rectangle(40,275,50,280);

rectangle(40,285,50,290);

rectangle(40,295,50,300);

rectangle(40,305,50,310);

rectangle(40,315,50,320);

rectangle(40,325,50,330);

rectangle(10,185,20,190);

rectangle(10,195,20,200);

rectangle(10,205,20,210);

rectangle(10,215,20,220);

rectangle(10,225,20,230);

rectangle(10,235,20,240);

rectangle(10,245,20,250);

rectangle(10,255,20,260);

rectangle(10,265,20,270);

rectangle(10,275,20,280);

rectangle(10,285,20,290);

rectangle(10,295,20,300);

rectangle(10,305,20,310);

rectangle(10,315,20,320);

rectangle(10,325,20,330);

rectangle(70,330,85,336);

rectangle(70,340,85,346);

rectangle(70,350,85,356);

rectangle(100,365,130,395);

rectangle(80,370,90,390);

rectangle(60,370,70,390);

rectangle(140,370,150,390);

rectangle(160,370,170,390);

rectangle(180,320,210,330);

rectangle(250,320,260,340);

rectangle(120,290,140,320);

line(120,300,140,300);

line(120,310,140,310);

rectangle(200,200,300,280);

rectangle(90,420,150,460);

line(0,20,0,450);

line(0,450,90,450);

line(0,20,620,20);

line(150,450,250,450);

line(350,450,380,450);

rectangle(380,300,400,450);

rectangle(400,305,410,310);

rectangle(400,315,410,320);

rectangle(400,325,410,330);

rectangle(400,335,410,340);

rectangle(400,345,410,350);

rectangle(400,355,410,360);

rectangle(400,365,410,370);

rectangle(400,375,410,380);

rectangle(400,385,410,390);

rectangle(400,395,410,400);

rectangle(400,405,410,410);

rectangle(400,415,410,420);

rectangle(400,425,410,430);

rectangle(400,435,410,440);

rectangle(370,305,380,310);

rectangle(370,315,380,320);

rectangle(370,325,380,330);

rectangle(370,335,380,340);

rectangle(370,345,380,350);

rectangle(370,355,380,360);

rectangle(370,365,380,370);

rectangle(370,375,380,380);

rectangle(370,385,380,390);

rectangle(370,395,380,400);

rectangle(370,405,380,410);

rectangle(370,415,380,420);

rectangle(370,425,380,430);

rectangle(370,435,380,440);

rectangle(250,420,350,470);

rectangle(270,430,330,460);

rectangle(280,390,290,400);

rectangle(300,390,310,400);

rectangle(320,390,330,400);

rectangle(560,380,630,430);

rectangle(570,300,630,340);

circle(180,430,10);

circle(180,430,4);

circle(530,420,20);

circle(530,420,8);

rectangle(210,400,230,410);

rectangle(460,380,500,470);

line(400,450,620,450);

line(620,450,620,430);

line(620,380,620,340);

line(460,430,470,430);

line(470,430,475,400);

line(475,400,480,400);

line(480,400,485,430);

line(485,430,500,430);

line(620,200,620,300);

line(620,20,620,155);

rectangle(570,155,630,200);

circle(500,100,10);

circle(500,100,4);

line(380,112,396,137);

line(381,112,397,137);

rectangle(420,170,510,260);

rectangle(470,300,475,310);

rectangle(485,300,490,310);

rectangle(500,300,505,310);

rectangle(450,300,460,315);

rectangle(515,300,535,325);

rectangle(380,200,400,205);

rectangle(380,190,400,195);

rectangle(380,180,400,185);

rectangle(380,210,400,215);

rectangle(380,220,400,225);

rectangle(380,230,400,235);

circle(590,250,20);

circle(590,250,8);

circle(590,120,15);

circle(590,120,5);

rectangle(580,70,600,75);

rectangle(550,70,570,75);

rectangle(520,70,540,75);

setfillstyle(SOLID\_FILL,GREEN); //GREEN COLOUR

floodfill(600,50,WHITE);

circle(387,124,15);

circle(387,124,16);

setfillstyle(SOLID\_FILL,RED); //RED COLOURS

floodfill(516,301,WHITE);

floodfill(451,301,WHITE);

setfillstyle(SOLID\_FILL,BROWN); //ORANGE COLOURS

floodfill(591,131,WHITE);

floodfill(501,106,WHITE);

floodfill(185,438,WHITE);

floodfill(271,440,WHITE);

floodfill(221,91,WHITE);

floodfill(65,380,WHITE);

floodfill(85,380,WHITE);

floodfill(165,380,WHITE);

floodfill(145,380,WHITE);

floodfill(31,156,WHITE);

floodfill(85,206,WHITE);

floodfill(121,91,WHITE);

floodfill(61,156,WHITE);

setfillstyle(SOLID\_FILL,YELLOW); //YELLOW COLOURS

floodfill(591,261,WHITE);

floodfill(591,121,WHITE);

floodfill(532,430,WHITE);

floodfill(501,101,WHITE);

floodfill(181,431,WHITE);

floodfill(77,68,WHITE);

floodfill(36,428,WHITE);

floodfill(35,160,WHITE);

floodfill(81,201,WHITE);

floodfill(51,146,WHITE);

floodfill(467,68,WHITE);

setfillstyle(SOLID\_FILL,WHITE); //WHITE COLOURS

floodfill(591,251,WHITE);

floodfill(531,421,WHITE);

floodfill(371,306,WHITE);

floodfill(371,316,WHITE);

floodfill(371,326,WHITE);

floodfill(371,336,WHITE);

floodfill(371,346,WHITE);

floodfill(371,356,WHITE);

floodfill(371,366,WHITE);

floodfill(371,376,WHITE);

floodfill(371,386,WHITE);

floodfill(371,396,WHITE);

floodfill(371,406,WHITE);

floodfill(371,416,WHITE);

floodfill(371,426,WHITE);

floodfill(371,436,WHITE);

floodfill(401,306,WHITE);

floodfill(401,316,WHITE);

floodfill(401,326,WHITE);

floodfill(401,336,WHITE);

floodfill(401,346,WHITE);

floodfill(401,356,WHITE);

floodfill(401,366,WHITE);

floodfill(401,376,WHITE);

floodfill(401,386,WHITE);

floodfill(401,396,WHITE);

floodfill(401,406,WHITE);

floodfill(401,416,WHITE);

floodfill(401,426,WHITE);

floodfill(401,436,WHITE);

floodfill(31,421,WHITE);

floodfill(71,61,WHITE);

floodfill(461,61,WHITE);

floodfill(41,186,WHITE);

floodfill(41,196,WHITE);

floodfill(41,206,WHITE);

floodfill(41,216,WHITE);

floodfill(41,226,WHITE);

floodfill(41,236,WHITE);

floodfill(41,246,WHITE);

floodfill(41,256,WHITE);

floodfill(41,266,WHITE);

floodfill(41,276,WHITE);

floodfill(41,286,WHITE);

floodfill(41,296,WHITE);

floodfill(41,306,WHITE);

floodfill(41,316,WHITE);

floodfill(41,326,WHITE);

floodfill(11,186,WHITE);

floodfill(11,196,WHITE);

floodfill(11,206,WHITE);

floodfill(11,216,WHITE);

floodfill(11,226,WHITE);

floodfill(11,236,WHITE);

floodfill(11,246,WHITE);

floodfill(11,256,WHITE);

floodfill(11,266,WHITE);

floodfill(11,276,WHITE);

floodfill(11,286,WHITE);

floodfill(11,296,WHITE);

floodfill(11,306,WHITE);

floodfill(11,316,WHITE);

floodfill(11,326,WHITE);

setfillstyle(SOLID\_FILL,7); //GREY COLOURS

floodfill(471,301,WHITE);

floodfill(486,301,WHITE);

floodfill(501,301,WHITE);

floodfill(381,181,WHITE);

floodfill(381,191,WHITE);

floodfill(381,201,WHITE);

floodfill(381,211,WHITE);

floodfill(381,221,WHITE);

floodfill(381,231,WHITE);

floodfill(561,390,WHITE);

floodfill(251,421,WHITE);

floodfill(571,301,WHITE);

floodfill(571,156,WHITE);

floodfill(130,105,WHITE);

floodfill(130,111,WHITE);

floodfill(90,105,WHITE);

floodfill(487,435,WHITE);

floodfill(21,91,WHITE);

floodfill(22,120,WHITE);

floodfill(71,331,WHITE);

floodfill(71,341,WHITE);

floodfill(71,351,WHITE);

floodfill(90,115,WHITE);

floodfill(121,315,WHITE);

floodfill(121,295,WHITE);

setfillstyle(SOLID\_FILL,8);

floodfill(521,71,WHITE);

floodfill(551,71,WHITE);

floodfill(581,71,WHITE);

floodfill(480,180,WHITE);

floodfill(285,395,WHITE);

floodfill(305,395,WHITE);

floodfill(325,395,WHITE);

floodfill(100,430,WHITE);

floodfill(30,200,WHITE);

floodfill(220,220,WHITE);

floodfill(390,350,WHITE);

outtextxy(160,90,"GPIO");

outtextxy(100,140,"Raspberry Pi Model B+ V1.2");

outtextxy(100,155,"C Raspberry Pi 2018");

outtextxy(30,94,"PWR");

outtextxy(30,120,"ACT");

outtextxy(60,240,"D");

outtextxy(60,250,"I");

outtextxy(60,260,"S");

outtextxy(60,270,"P");

outtextxy(60,280,"L");

outtextxy(60,290,"A");

outtextxy(60,300,"Y");

outtextxy(100,358,"||||");

outtextxy(100,397,"||||");

outtextxy(180,312,"||||");

outtextxy(180,332,"||||");

outtextxy(260,328,"\_");

outtextxy(260,317,"\_");

outtextxy(242,323,"\_");

outtextxy(220,220,"SAMSUNG");

outtextxy(210,230,"K4P3EQ-RGC2");

outtextxy(220,270,"GPQ699R");

outtextxy(110,430,"PWR");

outtextxy(115,440,"IN");

outtextxy(280,382,"|");

outtextxy(285,382,"|");

outtextxy(300,382,"|");

outtextxy(305,382,"|");

outtextxy(320,382,"|");

outtextxy(325,382,"|");

outtextxy(280,410,"H D M I");

outtextxy(213,392,"||");

outtextxy(217,412,"|");

outtextxy(415,320,"C");

outtextxy(415,335,"A");

outtextxy(415,350,"M");

outtextxy(415,365,"E");

outtextxy(415,380,"R");

outtextxy(415,395,"A");

outtextxy(430,440,"A/V");

outtextxy(552,437,"ETHERNET");

outtextxy(585,345,"USB");

outtextxy(585,205,"USB");

outtextxy(380,120,"Pb");

outtextxy(418,162,"||||||||||||");

outtextxy(418,262,"||||||||||||");

outtextxy(412,165,"\_");

outtextxy(412,170,"\_");

outtextxy(412,175,"\_");

outtextxy(412,180,"\_");

outtextxy(412,185,"\_");

outtextxy(412,190,"\_");

outtextxy(412,195,"\_");

outtextxy(412,200,"\_");

outtextxy(412,205,"\_");

outtextxy(412,210,"\_");

outtextxy(412,215,"\_");

outtextxy(412,220,"\_");

outtextxy(412,225,"\_");

outtextxy(412,230,"\_");

outtextxy(412,235,"\_");

outtextxy(412,240,"\_");

outtextxy(412,245,"\_");

outtextxy(412,250,"\_");

outtextxy(511,165,"\_");

outtextxy(511,170,"\_");

outtextxy(511,175,"\_");

outtextxy(511,180,"\_");

outtextxy(511,185,"\_");

outtextxy(511,190,"\_");

outtextxy(511,195,"\_");

outtextxy(511,200,"\_");

outtextxy(511,205,"\_");

outtextxy(511,210,"\_");

outtextxy(511,215,"\_");

outtextxy(511,220,"\_");

outtextxy(511,225,"\_");

outtextxy(511,230,"\_");

outtextxy(511,235,"\_");

outtextxy(511,240,"\_");

outtextxy(511,245,"\_");

outtextxy(511,250,"\_");

outtextxy(440,180,"S M S C");

outtextxy(425,200,"LAN954-JZX");

outtextxy(425,210,"B140BA1B17");

getch();

return 0;

}

laser()

{

int x1=50,y1=150,x2=50,y2=160,option;

getch();

cleardevice();

printf("Enter the number of the color:\n 1. BLUE\n 2. GREEN\n 3.CYAN \n 4. RED\n 5. MAGENTA \n 6. BROWN \n 7. LIGHTGRAY\n 8. DARKGRAY\n 9. LIGHTBLUE\n 10. LIGHTGREEN\n 11. LIGHTCYAN\n 12. LIGHTRED\n 13. LIGHTMAGENTA\n 14. YELLOW\n 15. WHITE\n");

scanf("%d",&option);

if (option==0 || option>15)

{

option = 4;

}

clrscr();

cleardevice();

ellipse(50,150,90,270,13,26);

ellipse(50,150,1,350,25,50);

line(50,124,250,124);

line(50,177,250,177);

ellipse(250,150,0,360,13,26);

ellipse(250,150,0,360,25,50);

rectangle(50,150,250,160);

setcolor(option);

while(!kbhit())

{

delay(2);

line(x1,y1,x2,y2);

x1+=1;

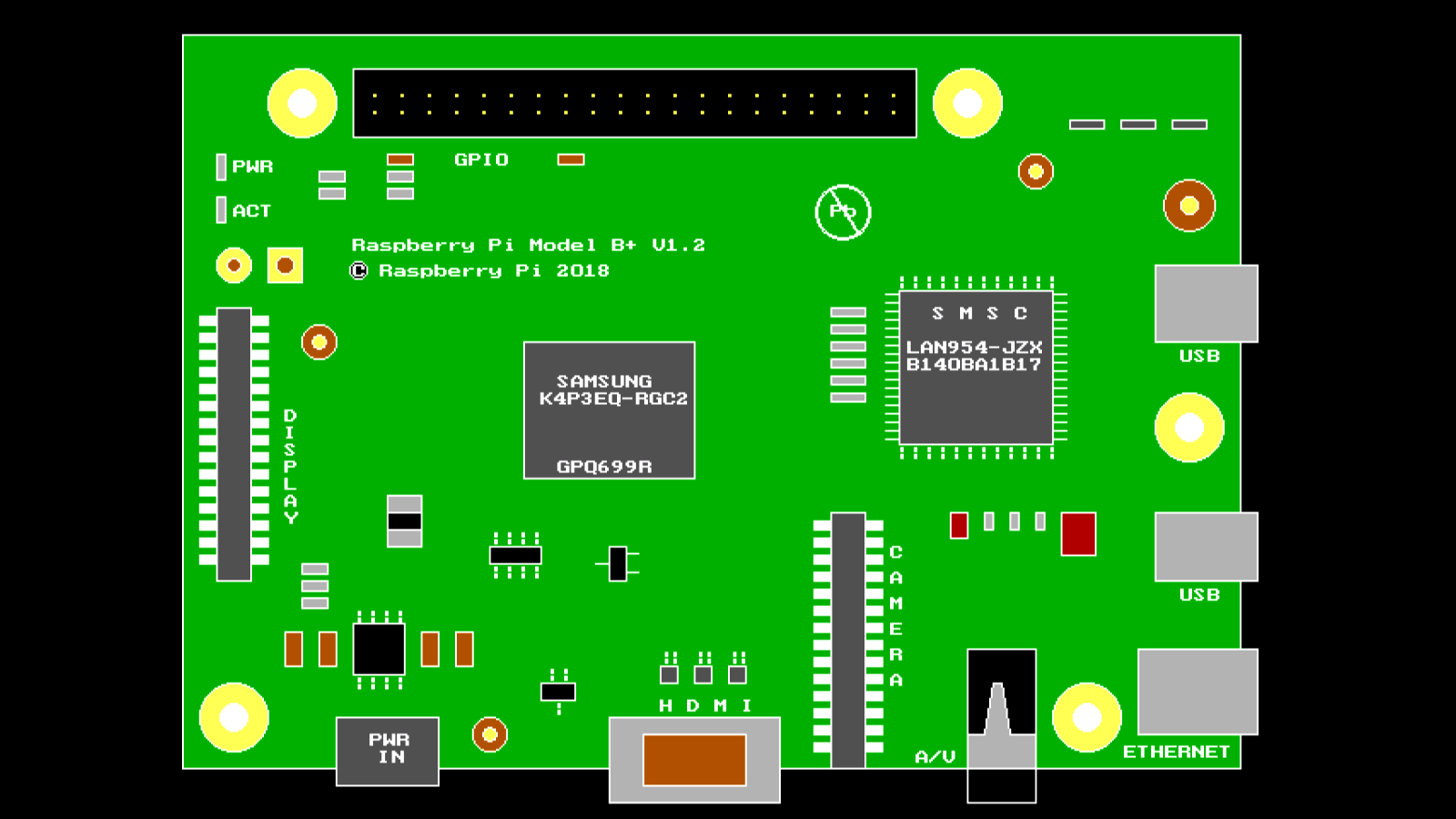
x2=x1;

}

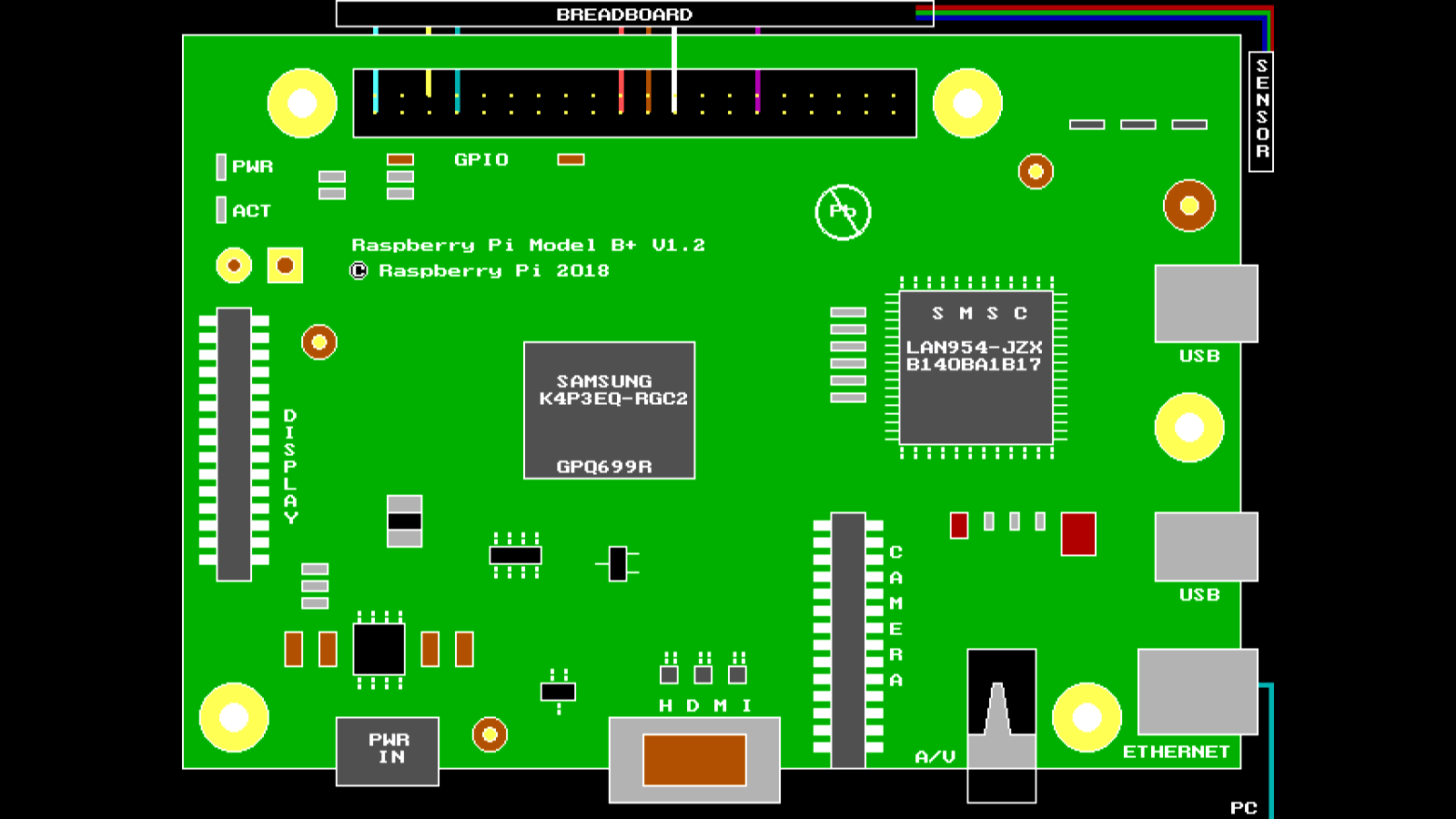
return 0;

}

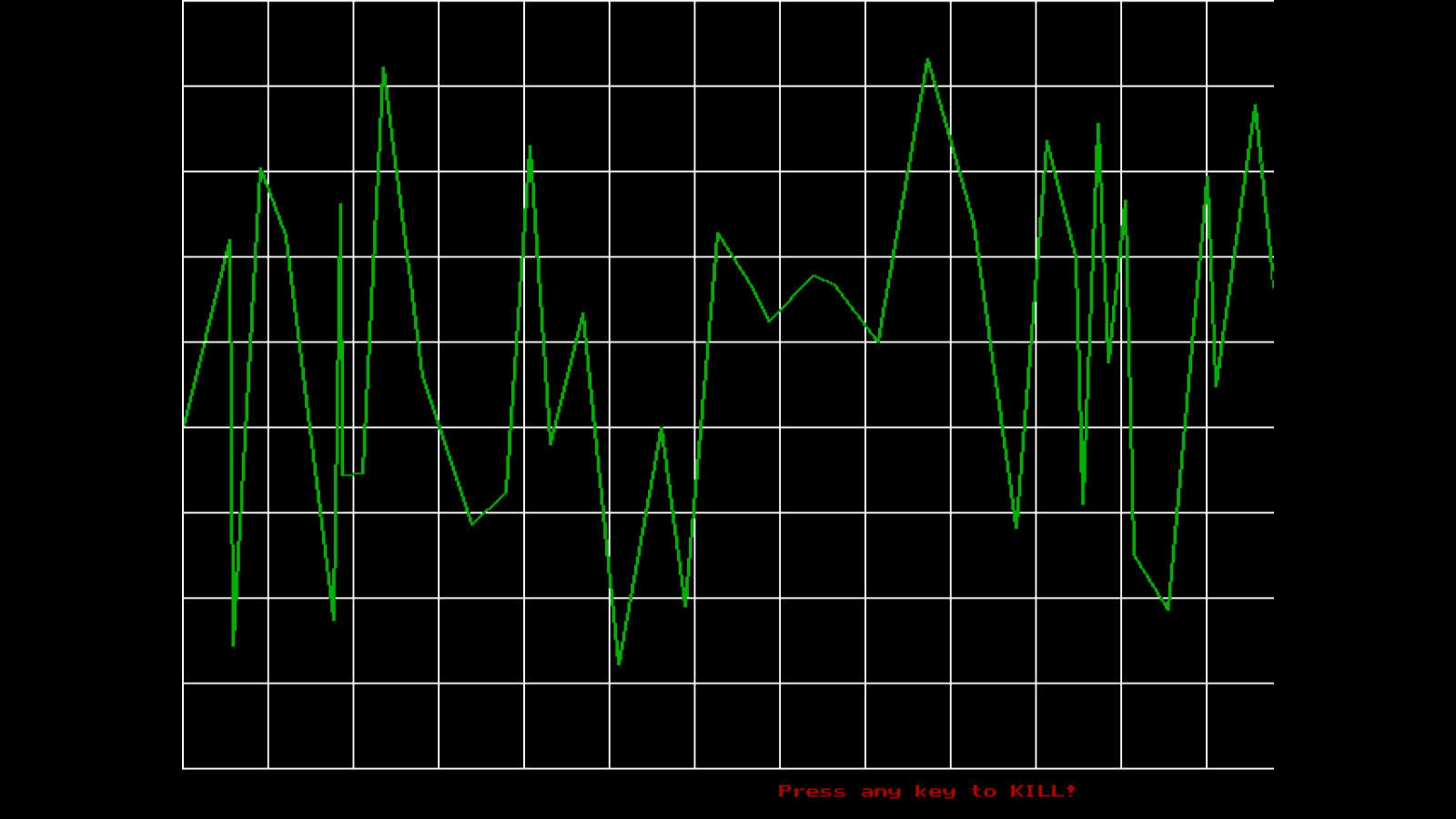
**OUTPUTS:**

****

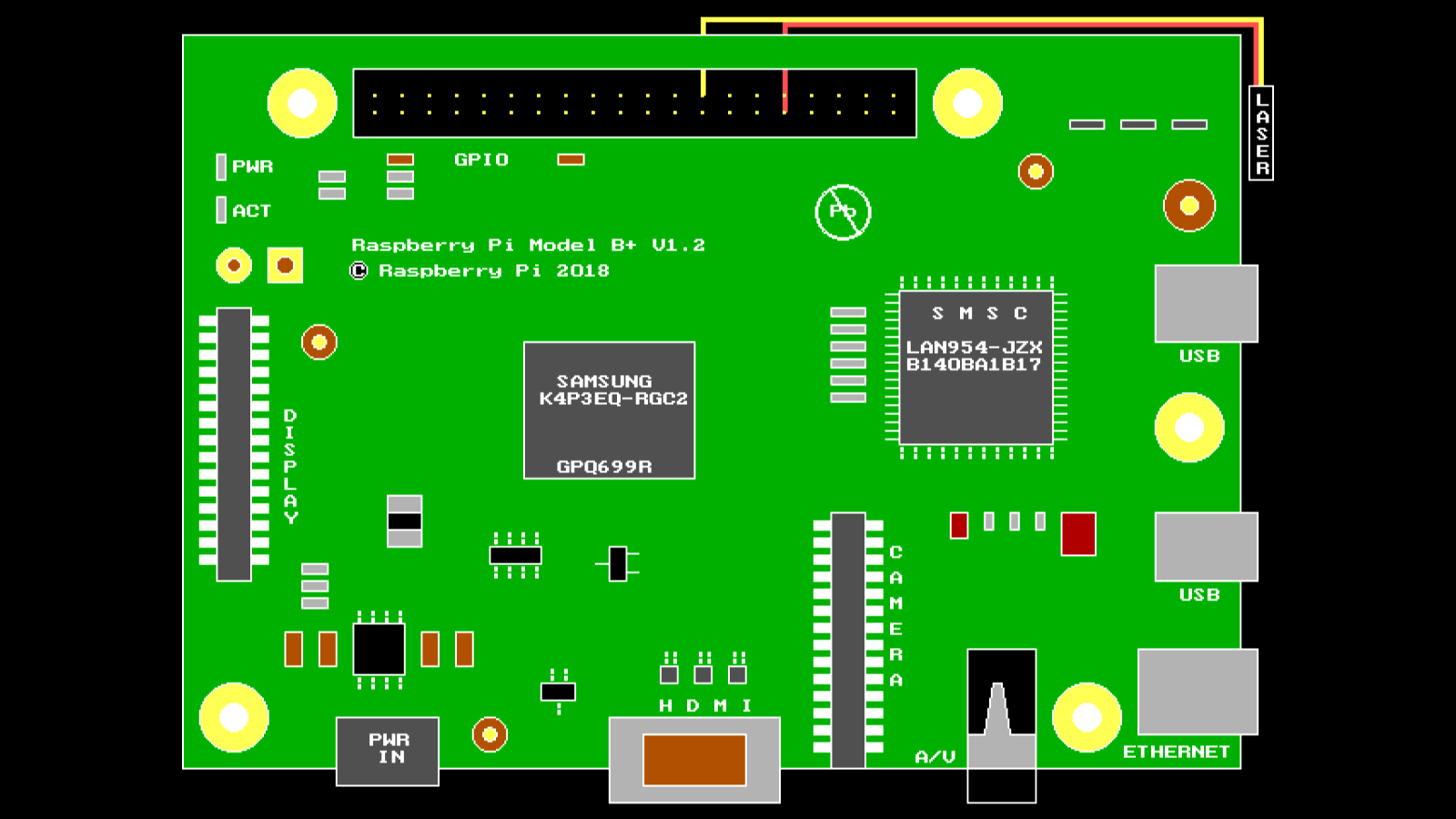
**RASPBERRY PI BOARD (UNCONNECTED)**

****

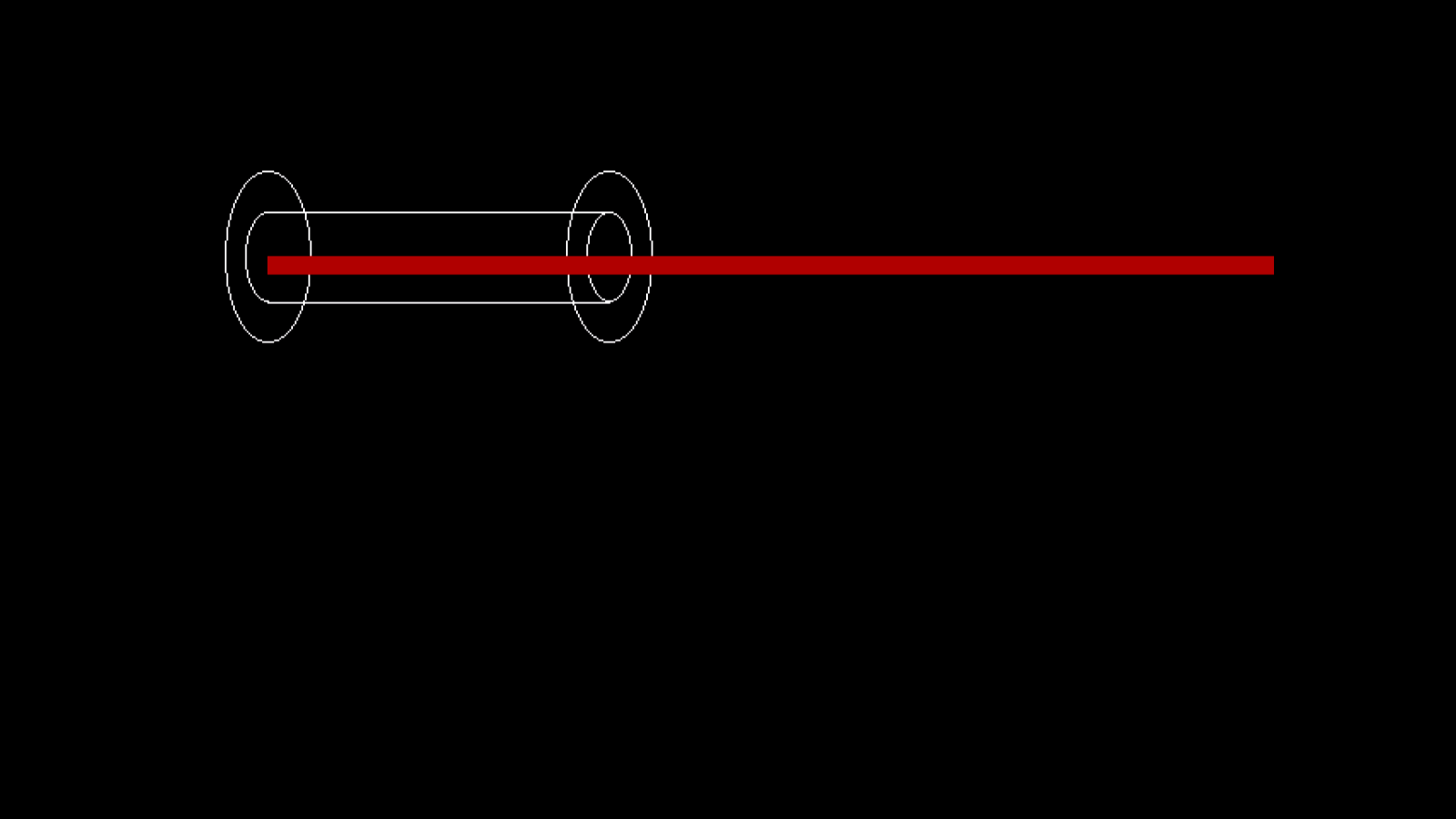
**CONNECTIONS FOR HEARTBEAT SENSOR**

****

**HEARTBEAT RATE GRAPH**

****

**CONNECTIONS FOR LASER**

****

**LASER BEAM OUTPUT**