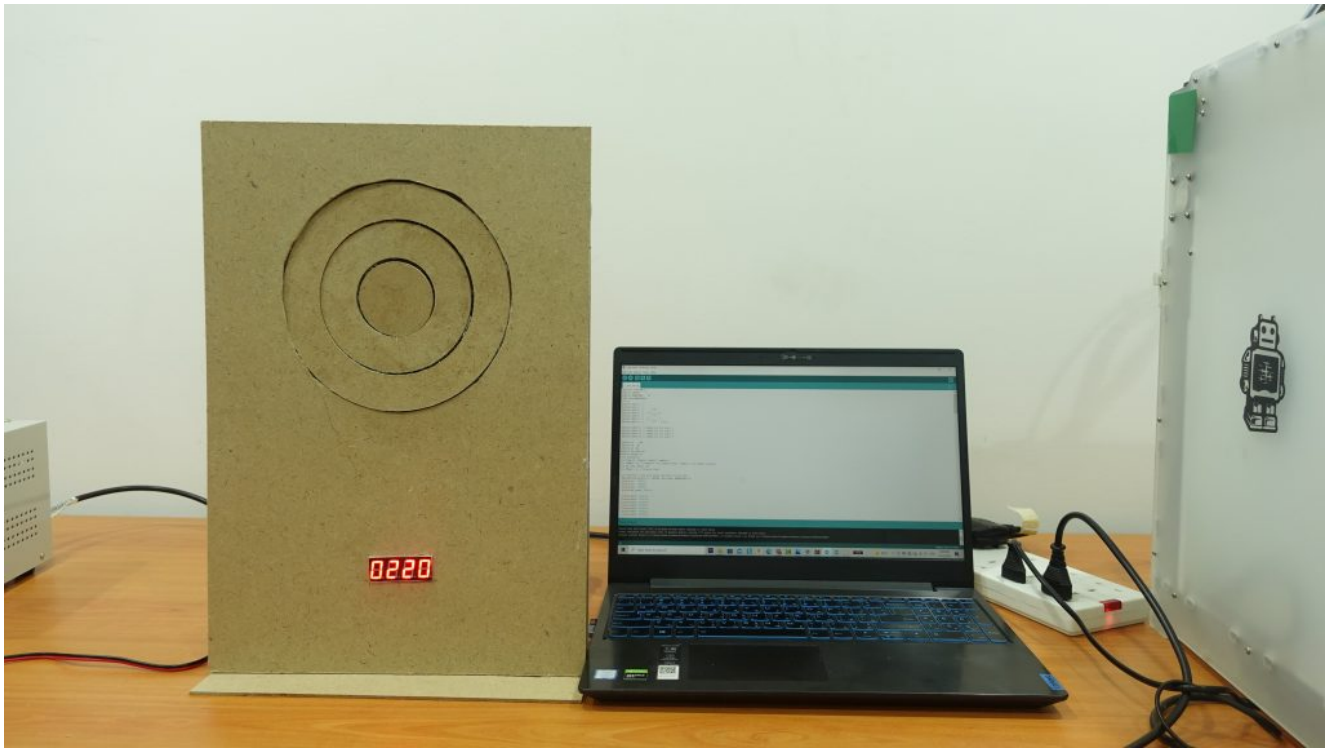
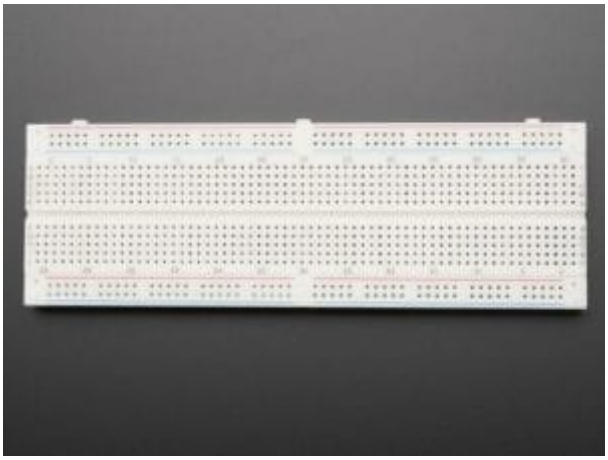
لعبة الرماية باستخدام الاردوينو

لعبة الرماية من الألعاب الرياضية المعروفة، ولها أشكال متعددة، في هذا الدرس سنتعلم كيف يمكننا، صنع نموذج مميز، و الكتروني من لعبة الرماية حيث يقوم اللاعب بالتصويب على الهدف، و يحتوي الهدف على ثلاثة مستويات، اذا كانت الاصابة قريبة من مركز الهدف تزداد النقاط، سنستخدم في هذا الدرس الاردوينو و شاشة سبعة مقاطع 4-أرقام لعرض النقاط.



المواد والأدوات

X1 اردوينو



X1 لوحة تجارب

X1 سبعة مقاطع 4–أرقام

مجموعة أسلاك توصيل(M/M)

X4 مقاومة 10 كيلو أوم

X1 ضغاط التحكم

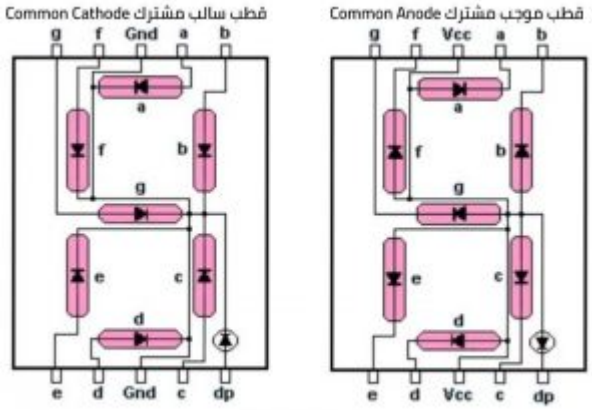
X1 سلك أردوينو

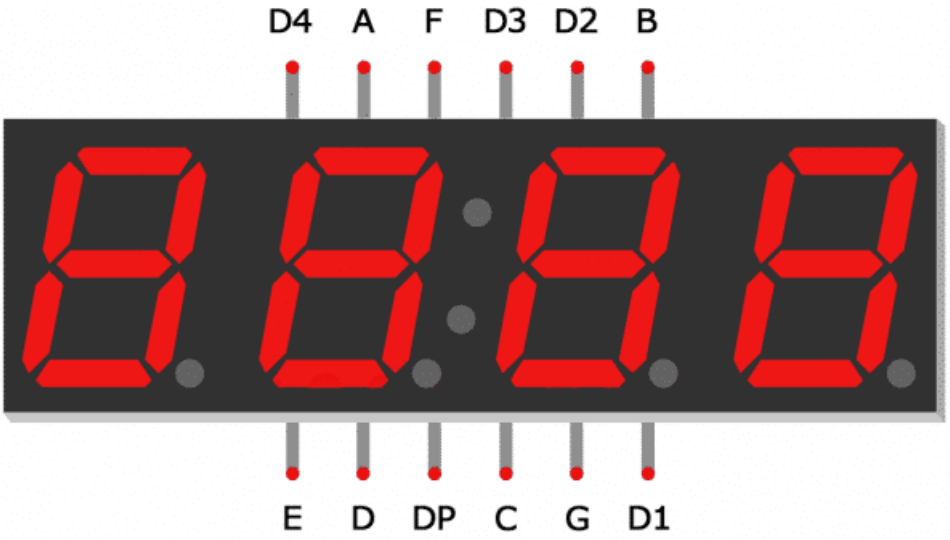
X10 زنبرك

وحدة سبع مقاطع – أربع ارقام

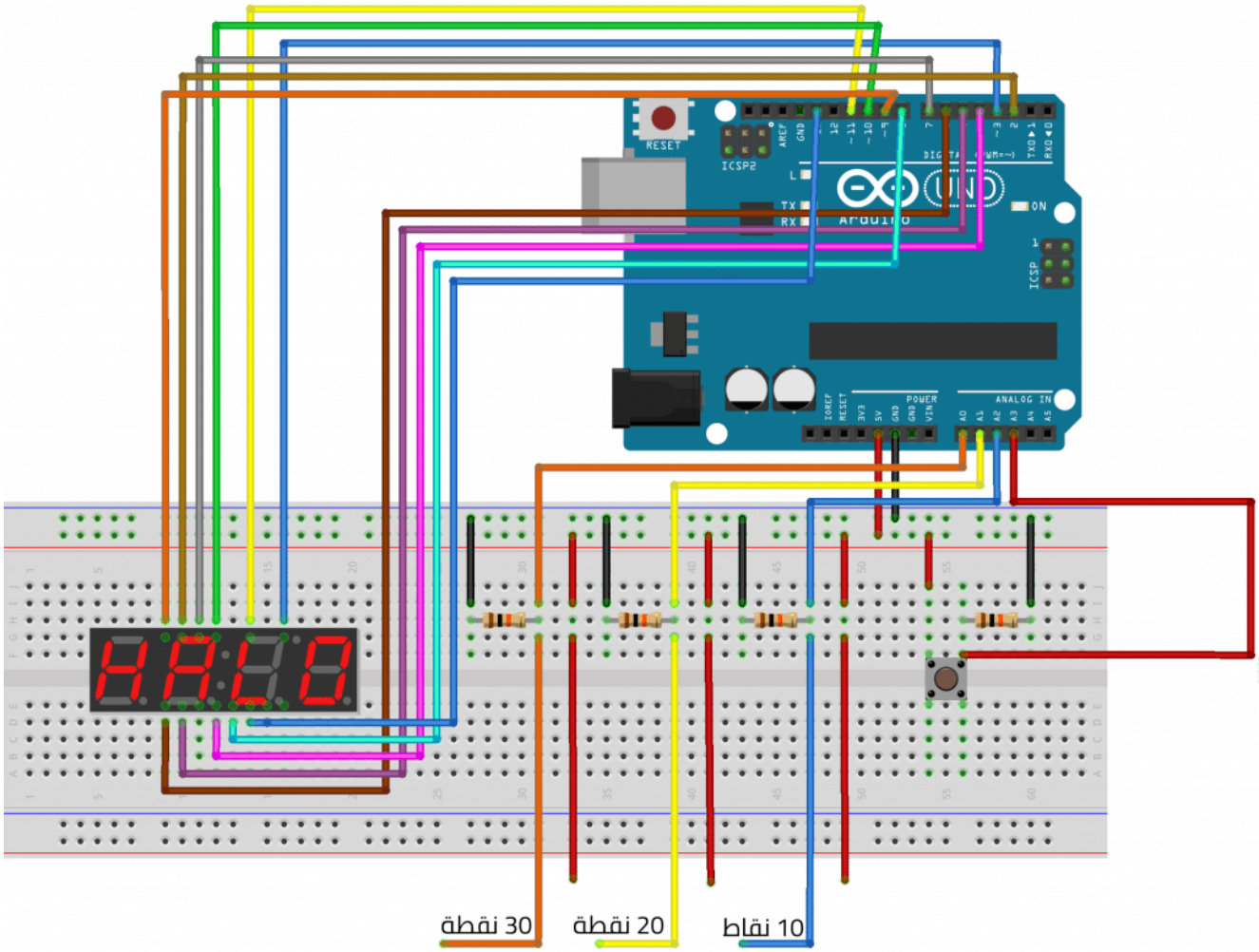
تستخدم وحدات العرض في كثير من المشاريع التي تتطلب عرض معلومات رقمية مثل درجة الحرارة، والوقت. يوجد شاشات عرض مكونة من 7 وحدات لتشكل رقم واحد تحتوي على 10 دبابيس، 7 منها للقطاعات، وواحد للنقطة، و2 دبابيس لمصدر الطاقة إما GND أو VCC حسب نوع الشاشة – قطب موجب مشترك Common Anode أو قطب سالب مشترك Common Cathode-
من ناحية أخرى، لا تحتوي الشاشة المكونة من 4 أرقام على 40 دبوس، بل تحتوي على 12 دبوس فقط: 7 للمقاطع وواحد للنقطة ودبابيس اختيار مكونة من 4 أرقام تحدد الرقم الذي سيكون نشطًا لعرض الحرف الذي أرسله Arduino على دبابيس قطعة 7 + 1.
يتم ترتيب المقاطع السبعة بطريقتين: إما أن يتم تجميع كل قطب سالب في دبوس واحد (GND)، وهذا النوع يسمى سالب مشترك Common Cathode، أو يتم تجميع كل قطب موجب في دبوس واحد (Vcc)، وهذا النوع يسمى قطب موجب مشترك Common Anode. وبالمثل، فإن الأجزاء المكونة من 7 أرقام والمكونة من 4 أرقام هي إما قطب موجب مشترك Common Anode أو قطب

سالب مشترك Common Cathode، وبالتالي توجد أربعة دبابيس مشتركة للأرقام الأربعة.





توصيل الدائرة



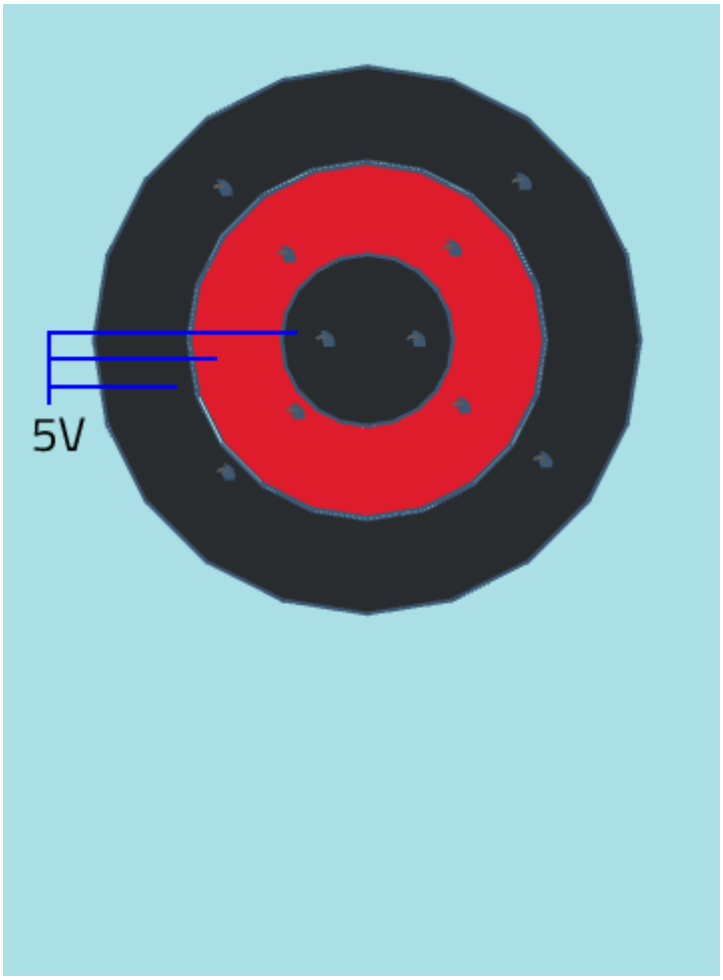
نهاية أطراف الأسلاك يتم توصيلها مع الحلقات في الهيكل بحيث تكون أصغر حلقة موصله مع A 0 والتي تليها مع A1 والأخيرة مع A2

الهيكل

فكرة المشروع وضع ثلاث حلقات مكونه من مادة موصلة على قاعدة، كل حلقة متصلة مع خط الطاقة 5 فولت بالاردوينو، تنفصل عنها 3 حلقات باستخدام الزنبرك موصلة مع منافذ الأردوينو

إذا استطاع اللاعب إصابة الهدف تتلامس الحلقات ويتم حساب النقاط ومن ثم عرضها على شاشة السبع وحدات أربع أرقام

في هذا الشكل يظهر توزيع الزنبركات والحلقات المتصلة على القاعدة الأساسية



اذا كنت تمتلك طابعة ثلاثية الأبعاد قم بطباعة أداة القوس و السهم التي ستستخدمها لتصويب الأهداف من خلال الرابط

الكود البرمجي

```
#define aPin 2
#define bPin 1
#define cPin 4
#define dPin 5
#define ePin 6
#define fPin 7
#define gPin 8

#define c1Pin 9 // Common pin for digit 1
#define c2Pin 10 // Common pin for digit 2
#define c3Pin 11 // Common pin for digit 3
#define c4Pin 12 // Common pin for digit 4

#define c1 A0
#define c2 A1
#define c3 A2
#define bt_reset A3
long Counter=0;
int flag1=0, flag2=0, flag3=0, timer=0;
```

```
int Common = 0; //<Common=1; for Common anode> <Common=0; for Common cathode>
int On, Off, Off_C;
int DTime = 4; // Display timer

void setup(){ // put your setup code here, to run once

pinMode(c1, INPUT);
pinMode(c2, INPUT);
pinMode(c3, INPUT);
pinMode(bt_reset, INPUT);

pinMode(aPin, OUTPUT);
pinMode(bPin, OUTPUT);
pinMode(cPin, OUTPUT);
pinMode(dPin, OUTPUT);
pinMode(ePin, OUTPUT);
pinMode(fPin, OUTPUT);
pinMode(gPin, OUTPUT);
pinMode(c1Pin, OUTPUT);
pinMode(c2Pin, OUTPUT);
pinMode(c3Pin, OUTPUT);
pinMode(c4Pin, OUTPUT);


if(Common==1){On=0, Off=1, Off_C=0;} //for Common anode
else{On=1, Off=0, Off_C=1;} //for Common cathode

// Indicate that system is ready
for (int i = 9; i >=0; i--) {
showNumber(i);
digitalWrite(c1Pin, Common);
digitalWrite(c2Pin, Common);
digitalWrite(c3Pin, Common);
digitalWrite(c4Pin, Common);
delay(1000);
}
}

void loop(){

if(digitalRead (c1) == 1){
if(flag1==0){ flag1=1;
Counter = Counter+30;
if(Counter>9999){Counter=0;}
}}
else{flag1=0;}

if(digitalRead (c2) == 1){
if(flag2==0){ flag2=1;
Counter = Counter+20;
if(Counter<0){Counter=9999;}
}
}else{flag2=0;}
```

```
if(digitalRead (c3) == 1){
if(flag2==0){ flag2=1;
Counter = Counter+10;
if(Counter<0){Counter=9999;}
}
}else{flag3=0;}

if(digitalRead (bt_reset) == 1){
Counter=0;
if(timer<200){timer=timer+1;}
if(timer==200){

}
}else{timer=0;}


showNumber((Counter/1000)%10);
digitalWrite(c1Pin, Common);
delay(DTime);
digitalWrite(c1Pin, Off_C);

showNumber((Counter/100)%10);
digitalWrite(c2Pin, Common);
delay(DTime);
digitalWrite(c2Pin, Off_C);

showNumber((Counter/10)%10);
digitalWrite(c3Pin, Common);
delay(DTime);
digitalWrite(c3Pin, Off_C);

showNumber(Counter%10);
digitalWrite(c4Pin, Common);
delay(DTime);
digitalWrite(c4Pin, Off_C);

}


void showNumber(int x){

switch(x){
case 1: one(); break;
case 2: two(); break;
case 3: three(); break;
case 4: four(); break;
case 5: five(); break;
case 6: six(); break;
case 7: seven(); break;
case 8: eight(); break;
case 9: nine(); break;
default: zero(); break;
}
```

```
}

void one(){
digitalWrite( aPin, Off); //
digitalWrite( bPin, On); // |
digitalWrite( cPin, On); // |
digitalWrite( dPin, Off); // |
digitalWrite( ePin, Off); // |
digitalWrite( fPin, Off);
digitalWrite( gPin, Off);
}

void two(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // |
digitalWrite( cPin, Off); // ____|
digitalWrite( dPin, On); // |
digitalWrite( ePin, On); // |____
digitalWrite( fPin, Off);
digitalWrite( gPin, On);
}

void three(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // |
digitalWrite( cPin, On); // ____|
digitalWrite( dPin, On); // |
digitalWrite( ePin, Off); // ____|
digitalWrite( fPin, Off);
digitalWrite( gPin, On);
}

void four(){
digitalWrite( aPin, Off); //
digitalWrite( bPin, On); // | |
digitalWrite( cPin, On); // |____|
digitalWrite( dPin, Off); // |
digitalWrite( ePin, Off); // |
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}

void five(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, Off); // |
digitalWrite( cPin, On); // |____
digitalWrite( dPin, On); // |
digitalWrite( ePin, Off); // ____|
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}

void six(){
```

```
digitalWrite( aPin, On); // ____
digitalWrite( bPin, Off); // |
digitalWrite( cPin, On); // |____
digitalWrite( dPin, On); // | |
digitalWrite( ePin, On); // |____|
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}

void seven(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // |
digitalWrite( cPin, On); // |
digitalWrite( dPin, Off); // |
digitalWrite( ePin, Off); // |
digitalWrite( fPin, Off);
digitalWrite( gPin, Off);
}

void eight(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // | |
digitalWrite( cPin, On); // |____|
digitalWrite( dPin, On); // | |
digitalWrite( ePin, On); // |____|
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}

void nine(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // | |
digitalWrite( cPin, On); // |____|
digitalWrite( dPin, On); // |
digitalWrite( ePin, Off); // ____|
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}

void zero(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // | |
digitalWrite( cPin, On); // | |
digitalWrite( dPin, On); // | |
digitalWrite( ePin, On); // |____|
digitalWrite( fPin, On);
digitalWrite( gPin, Off);
}
```

شرح الكود البرمجي

نعرف منافذ شاشة السبع مقاطع اربع أرقام

```
#define aPin 2
#define bPin 1
#define cPin 4
#define dPin 5
#define ePin 6
#define fPin 7
#define gPin 8
#define c1Pin 9 // Common pin for digit 1
#define c2Pin 10 // Common pin for digit 2
#define c3Pin 11 // Common pin for digit 3
#define c4Pin 12 // Common pin for digit 4
```

نعرف الخطوط الثلاثة الخاسة بحلقات هدف الرماية

```
#define c1 A0
#define c2 A1
#define c3 A2
```

نعرف زر تصفير حساب النقط

```
#define bt_reset A3
```

نعرف متغير لحساب النقاط باسم counter

```
long Counter=0;
```

نعرف متغير باسم common نعطيه قيمة 1 اذا كان شاشة السبع مقاطع اربع أرقاب ذات قطب موجب مشترك و 0 اذا كانت ذات قطب سابي مشترك

```
int flag1=0, flag2=0, flag3=0, timer=0;
int Common = 0; //<Common=1; for Common anode> <Common=0; for Common cathode>
int On, Off, Off_C;
int DTime = 4; // Display timer
```

في دالة void setup نعرف منافذ الدخل والخرج

```
void setup(){ // put your setup code here, to run once

pinMode(c1, INPUT);
pinMode(c2, INPUT);
pinMode(c3, INPUT);
pinMode(bt_reset, INPUT);

pinMode(aPin, OUTPUT);
pinMode(bPin, OUTPUT);
pinMode(cPin, OUTPUT);
pinMode(dPin, OUTPUT);
pinMode(ePin, OUTPUT);
pinMode(fPin, OUTPUT);
```

```
pinMode(gPin, OUTPUT);
pinMode(c1Pin, OUTPUT);
pinMode(c2Pin, OUTPUT);
pinMode(c3Pin, OUTPUT);
pinMode(c4Pin, OUTPUT);


if(Common==1){On=0, Off=1, Off_C=0;} //for Common anode
else{On=1, Off=0, Off_C=1;} //for Common cathode
```

في بداية تشغيل البرنامج و للتأكد من أن جميع وحدات شاشة العرض في جميع الأرقام تعمل بشكل صحيح يتم اظهار عد تنازلي على الشاشة من 9 الى 0 على كل رقم

```
for (int i = 9; i >=0; i--) {
showNumber(i);
digitalWrite(c1Pin, Common);
digitalWrite(c2Pin, Common);
digitalWrite(c3Pin, Common);
digitalWrite(c4Pin, Common);
delay(1000);
}
}
```

في بداية دالة voide loop نحدد الشرط اذا قرأ الأردوينو المنفذ الخاص بالدائرة الداخلية على أنه 1 يتم اضافة قيمة 30 نقطة لمتغير النقاط counter

```
void loop(){

if(digitalRead (c1) == 1){
if(flag1==0){ flag1=1;
Counter = Counter+30;
if(Counter>9999){Counter=0;}
}}
else{flag1=0;}
```

اذا قرأ الأردوينو اشارة 1 من المنفذ A1 سيتم اضافة 20 نقطة

```
if(digitalRead (c2) == 1){
if(flag2==0){ flag2=1;
Counter = Counter+20;
if(Counter<0){Counter=9999;}
}
```

اذا قرأ الأردوينو اشارة 1 من المنفذ A2 سيتم اضافة 10 نقطة

```
}else{flag2=0;}
if(digitalRead (c3) == 1){
if(flag2==0){ flag2=1;
Counter = Counter+10;
if(Counter<0){Counter=9999;}
```

```
}
}else{flag3=0;}
```

اذا قرأ الأردوينو اشارة 1 من المنفذ A3 تصفير عداد النقاط أي عند الضغط على ضغاط التحكم

```
if(digitalRead (bt_reset) == 1){
Counter=0;
if(timer<200){timer=timer+1;}
if(timer==200){

}
}else{timer=0;}
```

الدوال التالية لتعريف الأرقام من 0– 9

```


void one(){
digitalWrite( aPin, Off); //
digitalWrite( bPin, On); // |
digitalWrite( cPin, On); // |
digitalWrite( dPin, Off); // |
digitalWrite( ePin, Off); // |
digitalWrite( fPin, Off);
digitalWrite( gPin, Off);
}

void two(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // |
digitalWrite( cPin, Off); // ____|
digitalWrite( dPin, On); // |
digitalWrite( ePin, On); // |____
digitalWrite( fPin, Off);
digitalWrite( gPin, On);
}

void three(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // |
digitalWrite( cPin, On); // ____|
digitalWrite( dPin, On); // |
digitalWrite( ePin, Off); // ____|
digitalWrite( fPin, Off);
digitalWrite( gPin, On);
}

void four(){
digitalWrite( aPin, Off); //
digitalWrite( bPin, On); // | |
digitalWrite( cPin, On); // |____|
digitalWrite( dPin, Off); // |
digitalWrite( ePin, Off); // |
```

```
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}

void five(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, Off); // |
digitalWrite( cPin, On); // |____
digitalWrite( dPin, On); // |
digitalWrite( ePin, Off); // ____|
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}

void six(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, Off); // |
digitalWrite( cPin, On); // |____
digitalWrite( dPin, On); // | |
digitalWrite( ePin, On); // |____|
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}

void seven(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // |
digitalWrite( cPin, On); // |
digitalWrite( dPin, Off); // |
digitalWrite( ePin, Off); // |
digitalWrite( fPin, Off);
digitalWrite( gPin, Off);
}

void eight(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // | |
digitalWrite( cPin, On); // |____|
digitalWrite( dPin, On); // | |
digitalWrite( ePin, On); // |____|
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}

void nine(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // | |
digitalWrite( cPin, On); // |____|
digitalWrite( dPin, On); // |
digitalWrite( ePin, Off); // ____|
digitalWrite( fPin, On);
digitalWrite( gPin, On);
}
```

```
void zero(){
digitalWrite( aPin, On); // ____
digitalWrite( bPin, On); // | |
digitalWrite( cPin, On); // | |
digitalWrite( dPin, On); // | |
digitalWrite( ePin, On); // |____|
digitalWrite( fPin, On);
digitalWrite( gPin, Off);
}
```