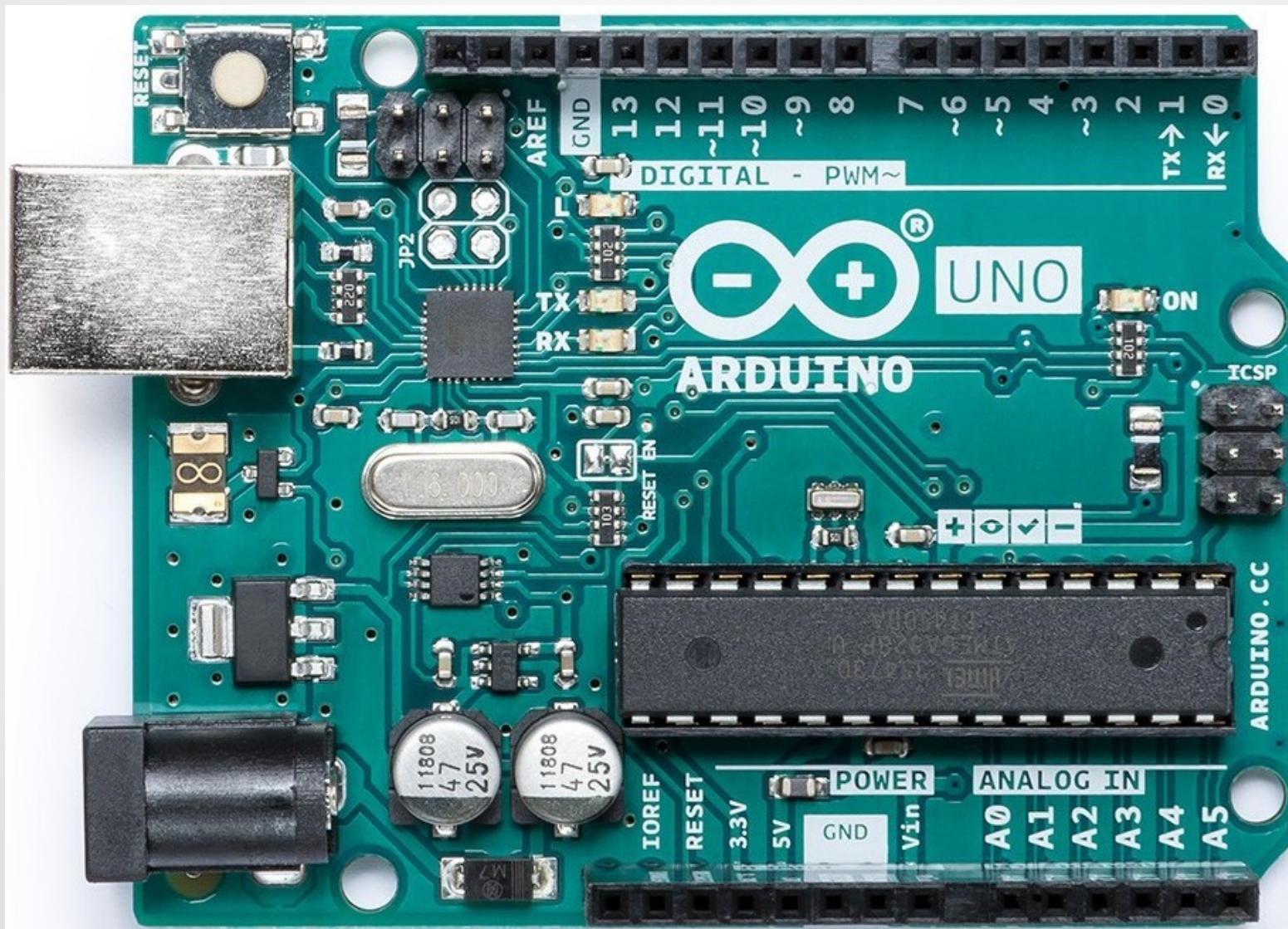


# Display de 7 segmentos





# duino

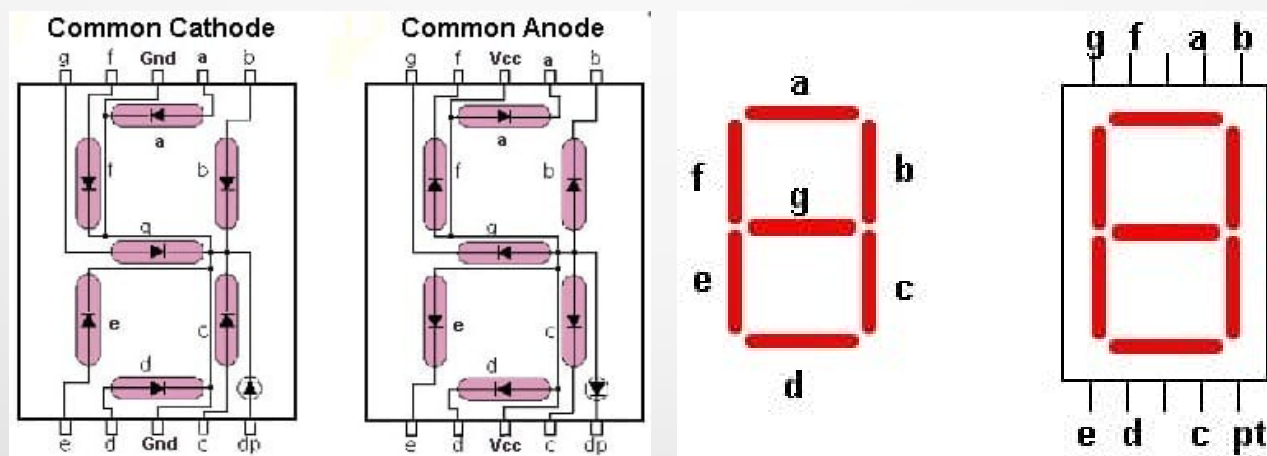


## Dixitais:

- Pins 0 a 13  
0 a 5V, 20 mA  
Low: 0 a 2V  
High: 3 a 5V
- ### **Entradas Analógicas**
- Pins A0 – A5  
0 a 5V, 20 mA prec 1024  
0 a 3V, 50 mA prec 1024
- ### **Saídas PWM**
- ~ Pins 3, 5, 6, 9, 10, 11  
0 a 5 V, 20 mA prec 256
- ### **Comunic. Serie TX/RX**
- Pins 0 e 1
- ### **ICSP**
- 6 pins para comunicarse directamente co proc. Atmega328
  - 6 pins para programar o USB

# Display de 7 segmentos

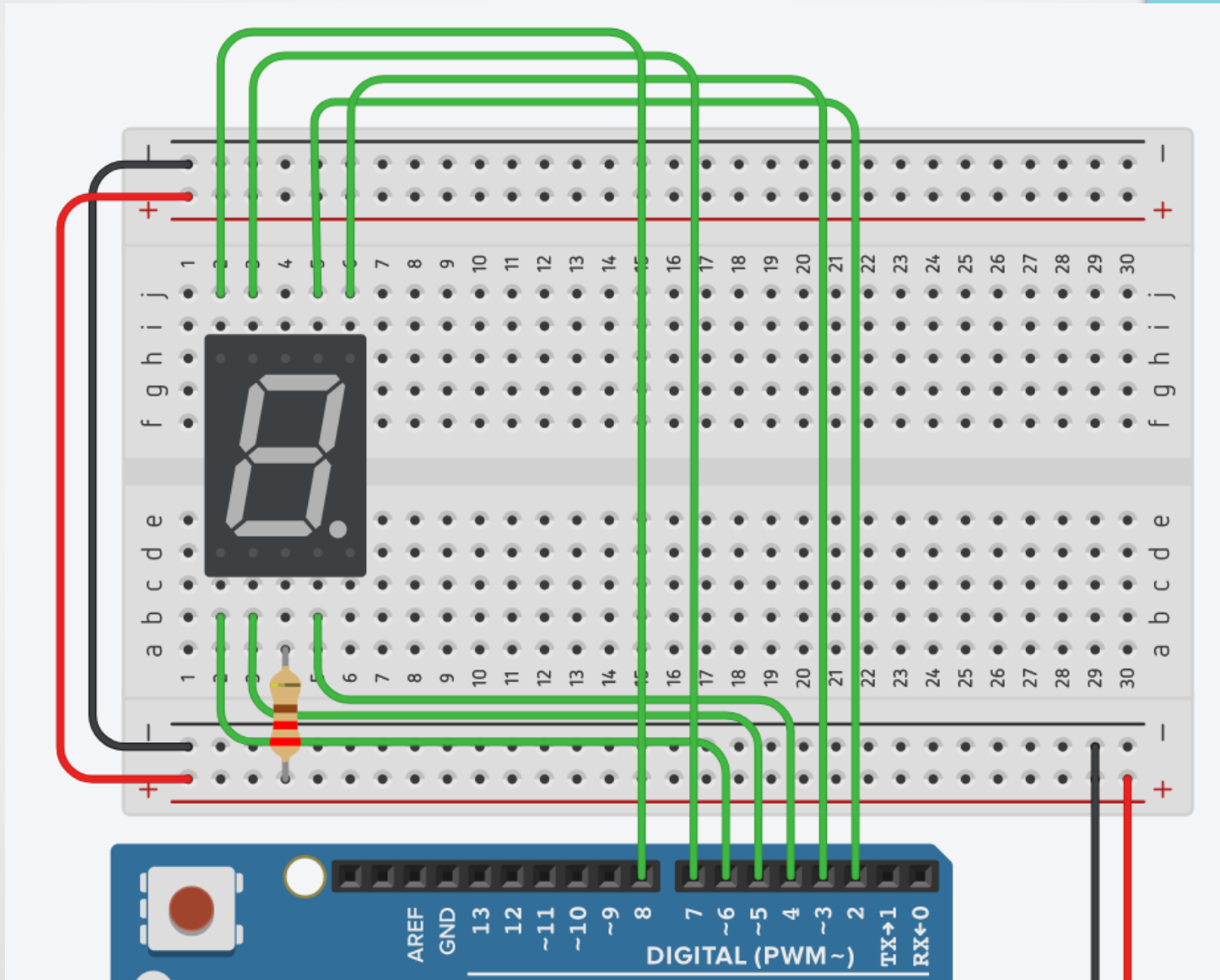
- Os displays de 7 segmentos son dispositivos que conteñen un array de 8 LEDs (sete segmentos máis un para o punto decimal). Con eles pódese representar caracteres alfanuméricos e o punto.
- Dependendo das conexións internas, poden ser de cátodo común (GND) ou de ánodo común (VCC). Para usalo conectaremos o común a unha resistencia de  $220\ \Omega$  (evitamos poñela en cada LED) e activamos os segmentos do cátodo común con HIGH e os do ánodo común con LOW.



# Display de 7 segmentos

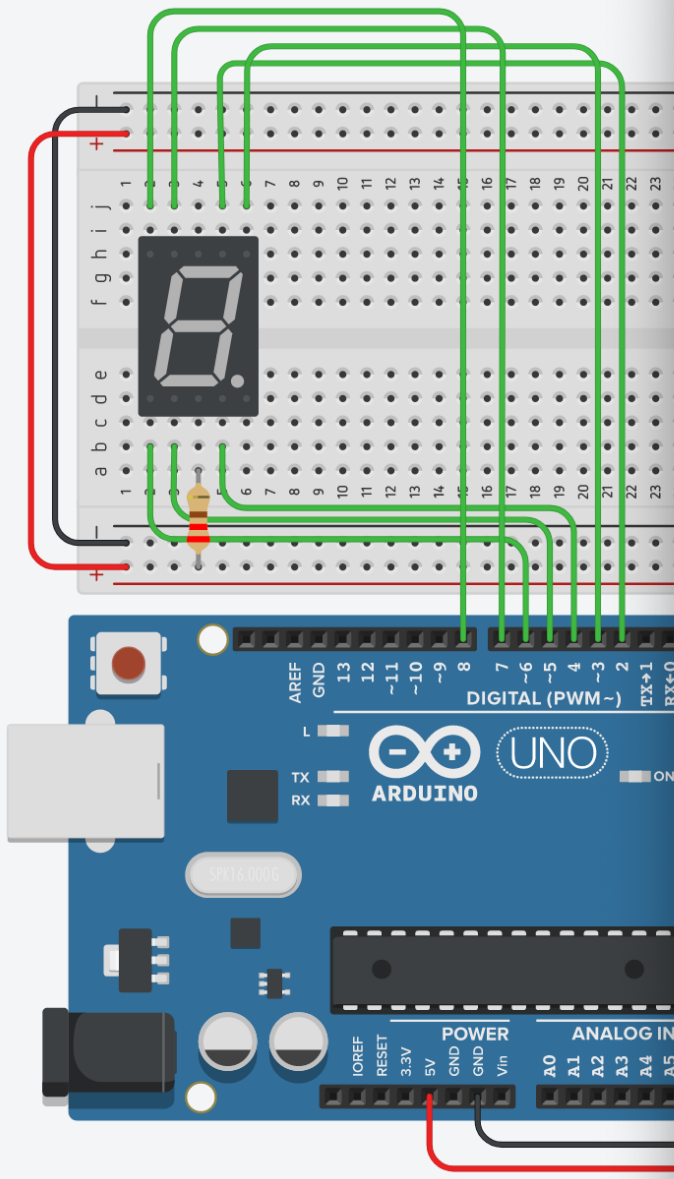
- En TinkerCAD podemos escoller se o display é de tipo ánodo ou cátodo común, no selector que aparece ao seleccionar o elemento.
- Imos facer que un display de 7 segmentos presente unha conta atrás de 9 a 0 en bucle.
- Usaremos un display en ánodo común e deixaremos o script preparado para que se poda usar cos de cátodo común de maneira fácil.
- Define e usa funcións que faciliten a lexibilidade do script.

# Display de 7 segmentos





# Display de 7 segmentos



## display.7.segmentos

```
/*
 * Realiza unha conta atrás de 9 a 0,
 * empregando un display de 7-segmentos
 * 5101AS (Cátodo común).
 *
 * Por ser cátodo común, temos que
 * conectar o pin común a terra (GND).
 * Se fose ánodo común, deberíamos
 * conectar o pin común a VCC (5 V).
 *
 * Os LEDs nomeanse con letras empezando
 * polo superior e seguindo no sentido
 * das agullas do reloxo: a, b, c, d, e, f, g
 * 0 ultimo é o g, que corresponde ao LED
 * central. Ademais, o h e o punto decimal.
 *
 * Se ponhemos numeros as conexions empezando
 * polo inferior esquerdo cara a dereita e
 * seguindo polo superior dereito cara a
 * esquerda:
 * 1 - e (inf esq)      6 - b (sup der)
 * 2 - d (inf cent)    7 - a (sup cent)
 * 3 - comun            8 - comun
 * 4 - c (inf der)     9 - f (sup esq)
 * 5 - dp (punto dec) 10 - g (central)
 */
```

```
//Descomentamos a seguinte linha se
//o display e de anodo comun.
```

```
//#define ANODO_COMUN
#ifdef ANODO_COMUN
#define HIGH 0
#define LOW 1
#endif
```

```
#define SUP 2
#define CENT 8
#define INF 5
#define SUPESQ 7
#define SUPDER 3
#define INFESQ 6
#define INFDER 4
```

## display.7.segmentos

```
#define SUP 2
#define CENT 8
#define INF 5
#define SUPESQ 7
#define SUPDER 3
#define INFESQ 6
#define INFDER 4

int contador = 10;
int tempo = 1000;

void setup() {
  for(int i=2; i<=8; i++) {
    pinMode(i, OUTPUT);
    digitalWrite(i, LOW);
  }
  Serial.begin(9600);
}

void loop() {
  apagaDisplay();
  contador--;
  if(contador == -1) {
    contador = 9;
  }
  Serial.print(contador);
  switch(contador) {
    case 0: cero(); break;
    case 1: un(); break;
    case 2: dous(); break;
    case 3: tres(); break;
    case 4: catro(); break;
    case 5: cinco(); break;
    case 6: seis(); break;
    case 7: sete(); break;
    case 8: oito(); break;
    case 9: nove(); break;
  }
  delay(tempo);
}
```

# Display de 7 segmentos

## display.7.segmentos

```
/*
 * Realiza unha conta atrás de 9 a 0,
 * empregando un display de 7-segmentos
 * 5101AS (Cátodo común).
 *
 * Por ser cátodo común, temos que
 * conectar o pin común a terra (GND).
 * Se fose ánodo común, deberíamos
 * conectar o pin común a VCC (5 V).
 *
 * Os LEDs nomeanse con letras empezando
 * polo superior e seguindo no sentido
 * das agullas do reloxo: a, b, c, d, e, f, g
 * O último é o g, que corresponde ao LED
 * central. Ademais, o h é o punto decimal.
 *
 * Se ponhemos números as conexións empezando
 * polo inferior esquerdo cara a dereita e
 * seguindo polo superior dereito cara a
 * esquerda:
 * 1 - e (inf esq)      6 - b (sup der)
 * 2 - d (inf cent)    7 - a (sup cent)
 * 3 - comun           8 - comun
 * 4 - c (inf der)     9 - f (sup esq)
 * 5 - dp (punto dec) 10 - g (central)
 */
```

```
//Descomentamos a seguinte linha se
//o display é de ánodo común.
//#define ANODO_COMUN
#ifdef ANODO_COMUN
    #define HIGH 0
    #define LOW 1
#endif
```

```
#define SUP      2
#define CENT     8
#define INF      5
#define SUPESQ   7
#define SUPDER   3
#define INFESQ   6
#define INFDER   4
```

## display.7.segmentos

```
#define SUP      2
#define CENT     8
#define INF      5
#define SUPESQ   7
#define SUPDER   3
#define INFESQ   6
#define INFDER   4

int contador = 10;
int tempo = 1000;

void setup() {
    for(int i=2; i<=8; i++) {
        pinMode(i, OUTPUT);
        digitalWrite(i, LOW);
    }
    Serial.begin(9600);
}

void loop() {
    apagaDisplay();
    contador--;
    if(contador == -1) {
        contador = 9;
    }
    Serial.print(contador);
    switch(contador) {
        case 0: cero(); break;
        case 1: un(); break;
        case 2: dous(); break;
        case 3: tres(); break;
        case 4: catro(); break;
        case 5: cinco(); break;
        case 6: seis(); break;
        case 7: sete(); break;
        case 8: oito(); break;
        case 9: nove(); break;
    }
    delay(tempo);
}
```

## display.7.segmentos

```
void apagaDisplay() {
    for(int i=2; i<=8; i++) {
        digitalWrite(i, LOW);
    }
}

void cero() {
    digitalWrite(SUP, HIGH);
    digitalWrite(SUPDER, HIGH);
    digitalWrite(INFDER, HIGH);
    digitalWrite(INF, HIGH);
    digitalWrite(INFESQ, HIGH);
    digitalWrite(SUPESQ, HIGH);
}

void un() {
    digitalWrite(SUPDER, HIGH);
    digitalWrite(INFDER, HIGH);
}

void dous() {
    digitalWrite(SUP, HIGH);
    digitalWrite(SUPDER, HIGH);
    digitalWrite(CENT, HIGH);
    digitalWrite(INFESQ, HIGH);
    digitalWrite(INF, HIGH);
}

void tres() {
    digitalWrite(SUP, HIGH);
    digitalWrite(SUPDER, HIGH);
    digitalWrite(INFDER, HIGH);
    digitalWrite(INF, HIGH);
    digitalWrite(CENT, HIGH);
}

void catro() {
    digitalWrite(SUPESQ, HIGH);
    digitalWrite(SUPDER, HIGH);
    digitalWrite(INFDER, HIGH);
    digitalWrite(CENT, HIGH);
}
```

## display.7.segmentos

```
}

void cinco() {
    digitalWrite(SUP, HIGH);
    digitalWrite(SUPESQ, HIGH);
    digitalWrite(CENT, HIGH);
    digitalWrite(INFDER, HIGH);
    digitalWrite(INF, HIGH);
}

void seis() {
    digitalWrite(SUP, HIGH);
    digitalWrite(SUPDER, HIGH);
    digitalWrite(INFDER, HIGH);
    digitalWrite(INF, HIGH);
    digitalWrite(INFESQ, HIGH);
    digitalWrite(CENT, HIGH);
}

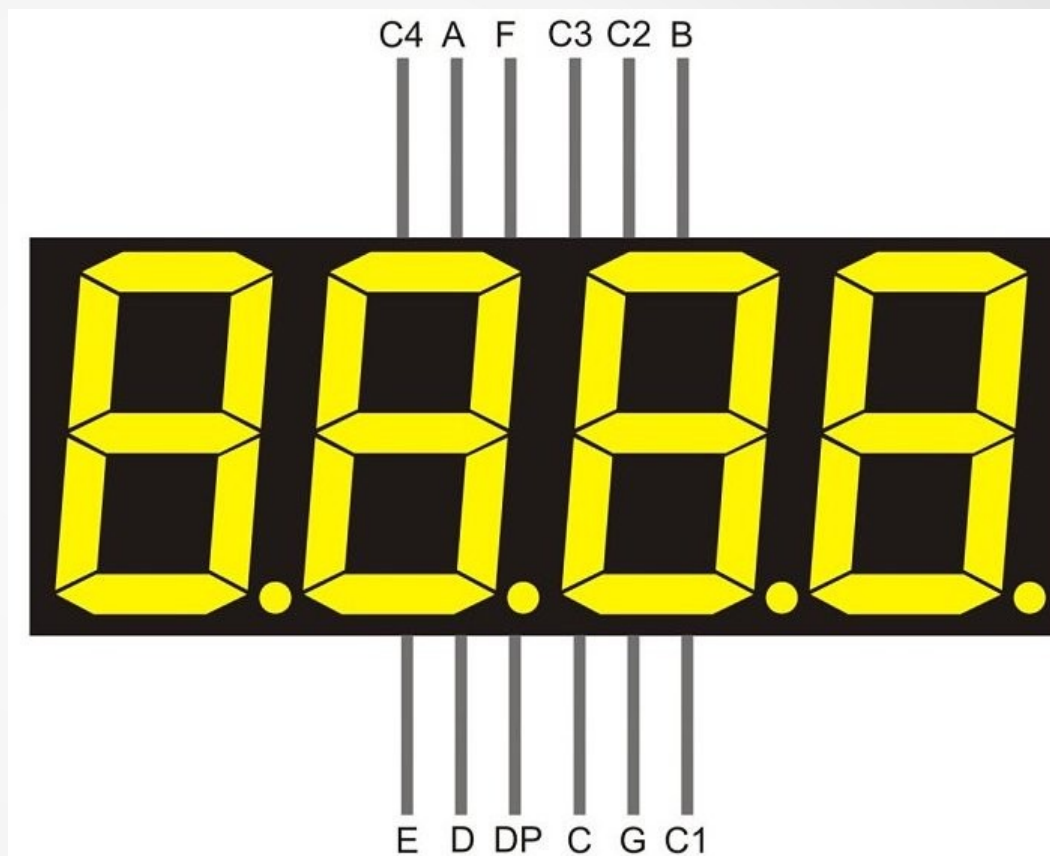
void sete() {
    digitalWrite(SUP, HIGH);
    digitalWrite(SUPDER, HIGH);
    digitalWrite(INFDER, HIGH);
}

void oito() {
    digitalWrite(SUP, HIGH);
    digitalWrite(SUPDER, HIGH);
    digitalWrite(INFDER, HIGH);
    digitalWrite(INF, HIGH);
    digitalWrite(INFESQ, HIGH);
    digitalWrite(SUPESQ, HIGH);
    digitalWrite(CENT, HIGH);
}

void nove() {
    digitalWrite(SUP, HIGH);
    digitalWrite(SUPESQ, HIGH);
    digitalWrite(SUPDER, HIGH);
    digitalWrite(INFDER, HIGH);
    digitalWrite(CENT, HIGH);
}
```

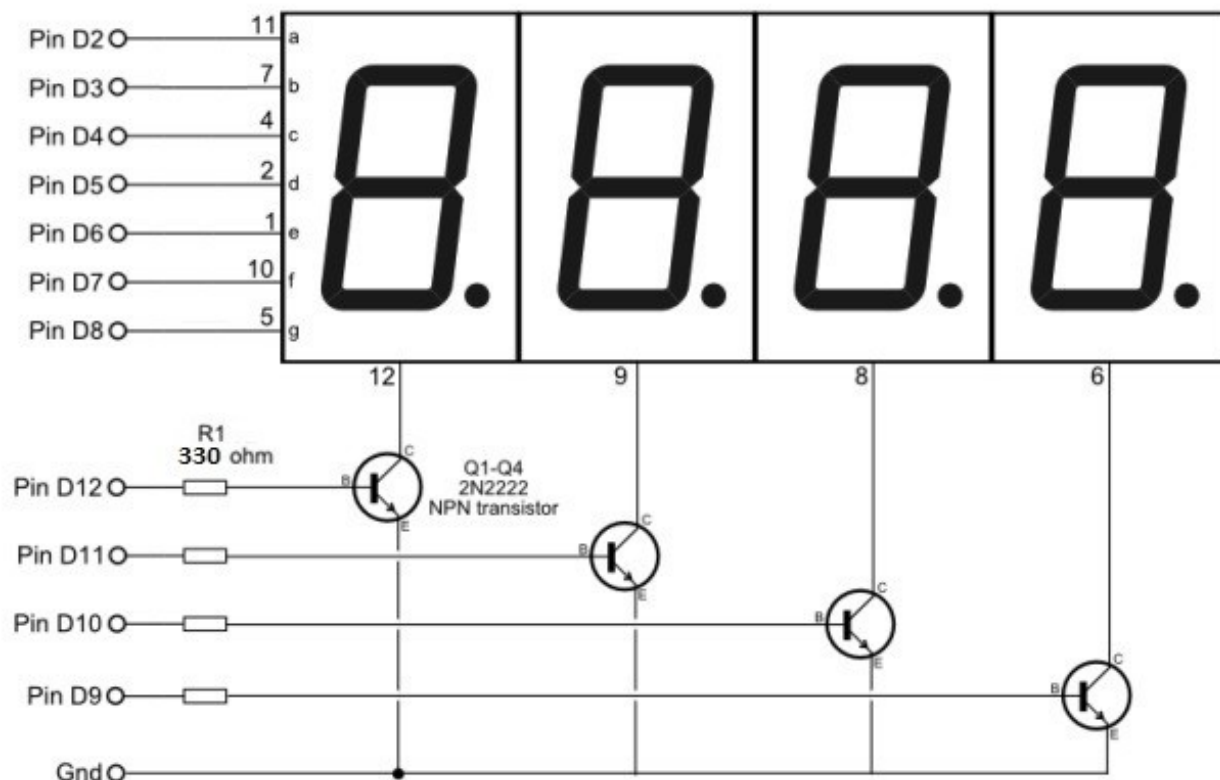
# Display de 7 segmentos 4 dígitos

- Un display de 7 segmentos e 4 dígitos é similar a un simple, en que dispoñemos de 8 pins para seleccionar os LEDs e ademais temos outros 4 pins para seleccionar cada un dos díxitos.
- Para usalos, temos que ter en conta que hai que habilitar o pin do dígito no que queremos escribir. Para que se manteña o texto, temos que ser suficientemente rápidos en escribir nos catro díxitos (uns poucos milisegundos) e proporcionar o tempo suficiente para que o usuario poda ler (varios centos de ms ou mellor varios segundos).





# Display de 7 segmentos 4 dígitos



- Un exemplo de montaxe é como o da figura adxunta.
- En calquera caso, convén comprobar a polaridade dos LEDs no noso compoñente concreto.

# Display de 7 segmentos 4 dígitos

```
mostra.milisegundos.display.7.dixitos
/*
 * Mostra nun display de 7 segmentos
 * e catro díxitos, o tempo que leva
 * aceso o Arduino.
 */

//Descomentamos a seguinte linha se
//o display e de anodo comun.
//#define ANODO_COMUN
#ifdef ANODO_COMUN
#define HIGH 0
#define LOW 1
#endif

//Declaramos os pins de cada número
//mediante un array de byte
byte dixito[10][8] = {
  {1,1,1,1,1,1,0,0}, // 0
  {0,1,1,0,0,0,0,0}, // 1
  {1,1,0,1,1,0,1,0}, // 2
  {1,1,1,1,0,0,1,0}, // 3
  {0,0,1,0,0,1,1,0}, // 4
  {1,0,1,1,0,1,1,0}, // 5
  {1,0,1,1,1,1,1,0}, // 6
  {1,1,1,0,0,0,0,0}, // 7
  {1,1,1,1,1,1,1,0}, // 8
  {1,1,1,0,0,1,1,0}, // 9
};

void setup() {
  for(int i=2; i<13; i++) {
    pinMode(i, OUTPUT);
  }
}

void loop() {
  //Conta ate 9999 segundos
  calcularDixitos(millis()/100);
}

void apagaDixitos() {
  for(int i=0; i<13; i++) {
    digitalWrite(i, LOW);
  }
}
```

```
mostra.milisegundos.display.7.dixitos
{1,1,1,1,1,1,0,0}, // 0
{1,1,1,0,0,1,1,0}, // 9
};

void setup() {
  for(int i=2; i<13; i++) {
    pinMode(i, OUTPUT);
  }
}

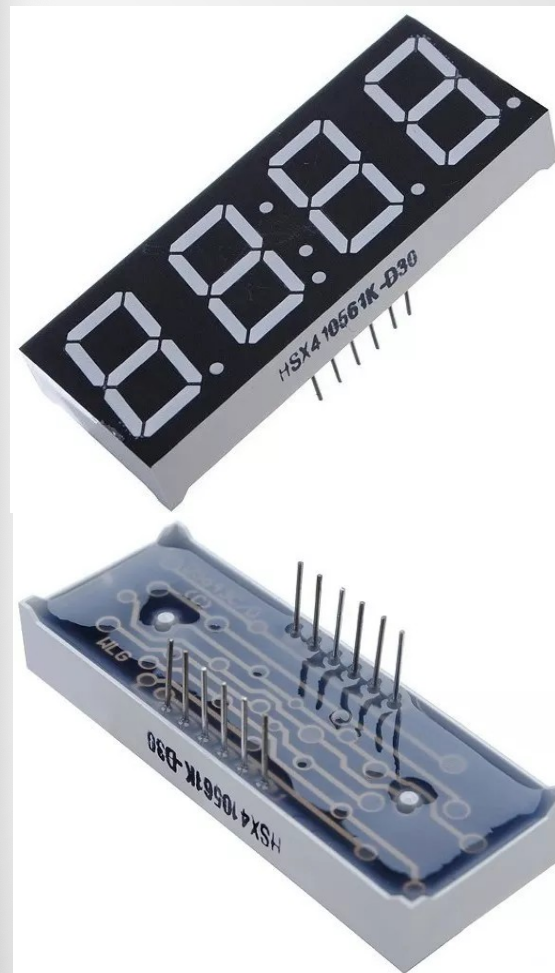
void loop() {
  //Conta ate 9999 segundos
  calcularDixitos(millis()/100);
}

void apagaDixitos() {
  for(int i=9; i<13; i++) {
    digitalWrite(i, LOW);
  }
}

void mostraDixito(int posDixito, int numero) {
  apagaDixitos();
  for(int i=0; i<8; i++) {
    digitalWrite(i+2, dixito[numero][i]);
  }
  //Activamos o dixito despois dos pins
  digitalWrite(posDixito + 9, HIGH);
}

void calcularDixitos(int numero) {
  int dixito0 = numero % 10;
  int dixito1 = (numero % 100) / 10;
  int dixito2 = (numero % 1000) / 100;
  int dixito3 = numero / 1000;

  mostraDixito(3, dixito3);
  mostraDixito(2, dixito2);
  mostraDixito(1, dixito1);
  mostraDixito(0, dixito0);
}
```



# Displays de 7 segmentos

- Nesta unidade aprendemos a:
  - usar sentencias do preprocesador para xeneralizar o script (`#ifdef` e `#endif`), neste caso para escoller entre ánodo e cátodo común,
  - declarar e usar functions definidas polo usuario,
  - cablear e usar displays de 7 segmentos (un e catro díxitos),
  - definir arrays de byte e int,
  - usar e percorrer arrays de byte e int, en combinación cos bucles `for()`.