

Pràctica 4: Coms

PROGRAAMCIÓ DE MICROPROCESADORS

Paula Dolado Aynié | Carles Gasia Romeu

Tabla de contenido

Coms	2
Activitat 1.1 Pujar volum i canviar de canal amb comandament	2
Codi	2
Explicació	7
Muntatge	8
Link	8
Activitat 1.2 Canvi de valors pel port sèrie	8
Codi	8
Explicació	15
Muntatge	15
Link	15

Coms

ACTIVITAT 1.1 PUJAR VOLUM I CANVIAR DE CANAL AMB COMANDAMENT

Codi

```
#include <IRremote.h>
int led1 = 5;
int led2 = 6;
int led3 = 7;
int led4 = 8;
int led5 = 9;
int codi;
int canal;
int volum = 120; //el volum ser  per defecte la meitat de 250
int canalAnterior=0;

void setup() {
  Serial.begin(9600);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  IrReceiver.begin(4);
}

void loop() {
  codi = readInfrared();
  if (codi != -1 ) {
    canalAnterior = canal;
    canal = decodeIR();
    if (canal != -1 ) {
      if(canal == 13){ //valor de canal menys
        canalAnterior--;
        canal = canalAnterior;
        if (canal <= 0) {
          canal = 0;
        }
      }
      if(canal == 11){ //valor de canal m s
        canalAnterior++;
        canal = canalAnterior;
        if (canal >= 9) {
          canal = 9;
        }
      }
      if(canal == 12){ //menys volum
        volum = volum - 10;
        canal = canalAnterior;
        if (volum <= 0) {
          volum = 0;
        }
      }
    }
  }
}
```

```

    }
}
if(canal == 10){ //més volum
    volum = volum + 10;
    canal = canalAnterior;
    if (volum >= 250) {
        volum = 250;
    }
}
}
LedOnIR();
Serial.print("Canal seleccionat: ");
Serial.print(canal);
Serial.print(" amb un volum: ");
Serial.println(volum);
}
}
void LedOnIR() {
    switch(canal){
        case 0:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,LOW);
            digitalWrite(led4,LOW);
            digitalWrite(led5,LOW);
            break;
        case 1:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,LOW);
            digitalWrite(led4,LOW);
            digitalWrite(led5,HIGH);
            break;
        case 2:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,LOW);
            digitalWrite(led4,HIGH);
            digitalWrite(led5,LOW);
            delay(250);
            break;
        case 3:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,LOW);
            digitalWrite(led4,HIGH);
            digitalWrite(led5,HIGH);
            break;
        case 4:
            analogWrite(led1, LedVolumIR());

```

```

        digitalWrite(led2,LOW);
        digitalWrite(led3,HIGH);
        digitalWrite(led4,LOW);
        digitalWrite(led5,LOW);
        delay(250);
        break;
    case 5:
        analogWrite(led1, LedVolumIR());
        digitalWrite(led2,LOW);
        digitalWrite(led3,HIGH);
        digitalWrite(led4,LOW);
        digitalWrite(led5,HIGH);
        delay(250);
        break;
    case 6:
        analogWrite(led1, LedVolumIR());
        digitalWrite(led2,LOW);
        digitalWrite(led3,HIGH);
        digitalWrite(led4,HIGH);
        digitalWrite(led5,LOW);
        delay(250);
        break;
    case 7:
        analogWrite(led1, LedVolumIR());
        digitalWrite(led2,LOW);
        digitalWrite(led3,HIGH);
        digitalWrite(led4,HIGH);
        digitalWrite(led5,HIGH);
        delay(250);
        break;
    case 8:
        analogWrite(led1, LedVolumIR());
        digitalWrite(led2,HIGH);
        digitalWrite(led3,LOW);
        digitalWrite(led4,LOW);
        digitalWrite(led5,LOW);
        delay(250);
        break;
    case 9:
        analogWrite(led1, LedVolumIR());
        digitalWrite(led2,HIGH);
        digitalWrite(led3,LOW);
        digitalWrite(led4,LOW);
        digitalWrite(led5,HIGH);
        delay(250);
        break;
    }
}
int LedVolumIR() {
    int pwm;

```

```

if (volum >= 0 && volum <= 25){
    pwm = 0;
}else if (volum > 25 && volum <= 50){
    pwm = 25;
}else if (volum > 50 && volum <= 75){
    pwm = 50;
}else if (volum > 75 && volum <= 100){
    pwm = 75;
}else if (volum > 100 && volum <= 125){
    pwm = 120;
}else if (volum > 125 && volum <= 150){
    pwm = 150;
}else if (volum > 150 && volum <= 175){
    pwm = 175;
}else if (volum > 175 && volum <= 200){
    pwm = 200;
}else if (volum > 200 && volum <= 225){
    pwm = 225;
}else if (volum > 225 && volum <= 250){
    pwm = 250;
}
return pwm;
}

int decodeIR() {
    int valorcanal = -1;
    switch(codi){
        case 1:
            valorcanal = 10; //més volum
            break;
        case 9:
            valorcanal = 12; //menys volum
            break;
        case 8:
            valorcanal = 13; //canal menys
            break;
        case 10:
            valorcanal = 11; //canal més
            break;
        case 12:
            valorcanal = 0; //canal 0
            break;
        case 16:
            valorcanal = 1; //canal 1
            break;
        case 17:
            valorcanal = 2; //canal 2
            break;
        case 18:
            valorcanal = 3; //canal 3
            break;
    }
}

```

```

    case 20:
        valorcanal = 4; //canal 4
        break;
    case 21:
        valorcanal = 5; //canal 5
        break;
    case 22:
        valorcanal = 6; //canal 6
        break;
    case 24:
        valorcanal = 7; //canal 7
        break;
    case 25:
        valorcanal = 8; //canal 8
        break;
    case 26:
        valorcanal = 9; //canal 9
        break;
    }
    return valorcanal;
}

int volumIR() {
    int valorcanal = 0;
    switch(codi){
        case 1:
            valorcanal = 1;
            break;
        case 9:
            valorcanal = 0;
            break;
    }
    return valorcanal;
}

int mapCodeToButton(unsigned long code) {
    // Check for codes from this specific remote
    if ((code & 0x0000FFFF) == 0x0000BFF0) {
        // No longer need the lower 16 bits. Shift the code by 16
        // to make the rest easier.
        code >>= 16;
        // Check that the value and inverse bytes are complementary.
        if (((code >> 8) ^ (code & 0x00FF)) == 0x00FF) {
            return code & 0xFF;
        }
    }
    return -1;
}

int readInfrared() {
    int result = -1;
    // Check if we've received a new code
    if (IrReceiver.decode()) {

```

```

// Get the infrared code
unsigned long code = IrReceiver.decodedIRData.decodedRawData;
// Map it to a specific button on the remote
result = mapCodeToButton(code);
// Enable receiving of the next value
IrReceiver.resume();
}
return result;
}

```

Explicació

En primer lloc, podem observar la declaració d'on son connectats tots els LED juntament amb variables. Tots aquests LEDs els trobarem dins de la funció de setup() on els inicialitzarem com a output. Cal destacar que al utilitzar un control remot d'infrarojos haurem d'incloure la seva llibreria amb: #include <IRremote.h> i en la seva inicialització indicarem el port del receptor.

D'altra banda, hi ha dues funcions:

- mapCodeToButton(): que retorna el valor en cas de prémer un botó del comandament a distància en 8 bits i en cas contrari un -1.
- readInfrared(): cridarà a la funció anterior i retorna el valor anterior del botó.

Seguidament, pel que fa al bucle principal farem ús del mètode esmentat anteriorment per llegir el botó pres i el guardarem a la variable codi, aquesta sempre que sigui diferent a -1, es a dir, quan rebí que l'usuari ha accionat un botó cridarà a la funció decodeIR() que te com a finalitat retornar un int i crearà una còpia d'aquest valor. En cas de ser un canal farem la conversió del valor que reben del botó des de la variable codi al valor real, per exemple, el botó 0 rep el valor de 12 i amb aquest mètode obtindrem 0.

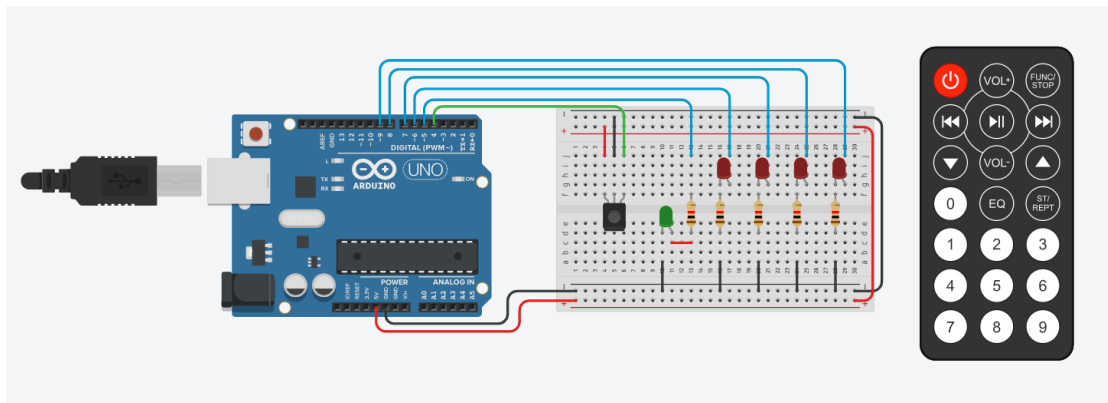
Dins d'aquest mètode podrem veure altres botons no numèrics que fan referència a pujar/baixar volum i pujar/baixar el número de canal.

- Cas1 Baixar un canal: quan la variable canal rebí el valor de 13 voldrem baixar un canal pel que restarem -1 a la còpia de canal i l'atorgarem a aquesta encara que quan detecti que el valor ja és 0 no disminuirà més
- Cas2 Pujar un canal: repetim l'acció anterior en sentir contrari, afegirem 1 al valor del canal i quan aquest sigui 9 ja no podrà augmentar-ne més.
- Cas3 Baixar el volum: quan la variable canal rebí el valor de 12 voldrem reduir el volum i per aquest cas s'utilitzarà la mateixa metodologia que el primer cas i mantindrem que el canal sigui l'anterior. Si no apliquéssim aquest canvi sempre apareixeria el valor de -1.
- Cas4 Pujar el volum: repetirem l'acció anterior fent una suma e indicant que el valor de volum mai podrà superar els 250.

Acabant amb el loop(), encendrem els LEDs e imprimirem per pantalla el canal i el volum actual.

El mètode de LedOnIR() tracta que per cada cas de canal, per cada numero indicat, encengui els LEDs corresponents. Com a diferencia veurem que el ledi es engegat de manera analògica degut a que volem que la seva intensitat variï depenent del volum. Per a que aquesta acció es realitzi cridarem a la funció LedVolumIR() que s'encarregarà de retornar el valor de la intensitat depenent del volum tenint en compte que el volum s'inicia a 120 prendrà aquesta intensitat i augmentarà o reduirà de manera proporcional.

Muntatge



Link

<https://www.tinkercad.com/things/36ZmBuQJ54O-practica4u/editel?returnTo=%2Fdashboard&sharecode=lcmC6FyrxLk54pQGkcQpqzaA3ktlt4PshP1bvtmJmNY>

ACTIVITAT 1.2 CANVI DE VALORS PEL PORT SÈRIE

Codi

```
#include <IRremote.h>
```

```

int led1 = 7;
int led2 = 5;
int led3 = 4;
int led4 = 3;
int led5 = 2;
int codi;
int canal;
int volum = 120; //el volum ser  per defecte la meitat de 250
int canalAnterior=0;
char received ='o';
void setup() {
  Serial.begin(9600);
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
  IrReceiver.begin(6);
  Serial.print("Que vols fer:");
  Serial.println();
}

void loop() {
  codi = readInfrared();
  if(Serial.available()){
    received = Serial.read();
    codi = serialtoinf(received);
  }
  if (codi != -1 ) {
    canalAnterior = canal;
    canal = decodeIR();
    if (canal != -1 ) {
      if(canal == 13){ //valor de canal menys
        canalAnterior--;
        canal = canalAnterior;
        if (canal <= 0) {
          canal = 0;
        }
      }
      if(canal == 11){ //valor de canal m s
        canalAnterior++;
        canal = canalAnterior;
        if (canal >= 9) {
          canal = 9;
        }
      }
      if(canal == 12){ //menys volum
        volum = volum - 10;
        canal = canalAnterior;
        if (volum <= 0) {

```

```

        volum = 0;
    }
}
if(canal == 10){ //més volum
    volum = volum + 10;
    canal = canalAnterior;
    if (volum >= 250) {
        volum = 250;
    }
}
}
LedOnIR();
Serial.print("Canal seleccionat: ");
Serial.print(canal);
Serial.print(" amb un volum: ");
Serial.println(volum);
Serial.print("Que vols fer:");
    Serial.println();
    delay(300);
}
}
int serialtoinf(char received){
    switch(received){
        case '0':
            return 12;
        case '1':
            return 16;
        case '2':
            return 17;
        case '3':
            return 18;
        case '4':
            return 20;
        case '5':
            return 21;
        case '6':
            return 22;
        case '7':
            return 24;
        case '8':
            return 25;
        case '9':
            return 26;
        case '+':
            return 10;
        case '-':
            return 8;
        case 'A':
            return 1;
        case 'V':

```

```

        return 9;
    }
}

void LedOnIR() {
    switch(canal){
        case 0:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,LOW);
            digitalWrite(led4,LOW);
            digitalWrite(led5,LOW);
            break;
        case 1:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,LOW);
            digitalWrite(led4,LOW);
            digitalWrite(led5,HIGH);
            break;
        case 2:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,LOW);
            digitalWrite(led4,HIGH);
            digitalWrite(led5,LOW);
            delay(250);
            break;
        case 3:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,LOW);
            digitalWrite(led4,HIGH);
            digitalWrite(led5,HIGH);
            break;
        case 4:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,HIGH);
            digitalWrite(led4,LOW);
            digitalWrite(led5,LOW);
            delay(250);
            break;
        case 5:
            analogWrite(led1, LedVolumIR());
            digitalWrite(led2,LOW);
            digitalWrite(led3,HIGH);
            digitalWrite(led4,LOW);
            digitalWrite(led5,HIGH);

```

```

        delay(250);
        break;
    case 6:
        analogWrite(led1, LedVolumIR());
        digitalWrite(led2, LOW);
        digitalWrite(led3, HIGH);
        digitalWrite(led4, HIGH);
        digitalWrite(led5, LOW);
        delay(250);
        break;
    case 7:
        analogWrite(led1, LedVolumIR());
        digitalWrite(led2, LOW);
        digitalWrite(led3, HIGH);
        digitalWrite(led4, HIGH);
        digitalWrite(led5, HIGH);
        delay(250);
        break;
    case 8:
        analogWrite(led1, LedVolumIR());
        digitalWrite(led2, HIGH);
        digitalWrite(led3, LOW);
        digitalWrite(led4, LOW);
        digitalWrite(led5, LOW);
        delay(250);
        break;
    case 9:
        analogWrite(led1, LedVolumIR());
        digitalWrite(led2, HIGH);
        digitalWrite(led3, LOW);
        digitalWrite(led4, LOW);
        digitalWrite(led5, HIGH);
        delay(250);
        break;
    }
}

int LedVolumIR() {
    int pwm;
    if (volum >= 0 && volum <= 25){
        pwm = 0;
    }else if (volum > 25 && volum <= 50){
        pwm = 25;
    }else if (volum > 50 && volum <= 75){
        pwm = 50;
    }else if (volum > 75 && volum <= 100){
        pwm = 75;
    }else if (volum > 100 && volum <= 125){
        pwm = 120;
    }else if (volum > 125 && volum <= 150){
        pwm = 150;
    }
}

```

```

}else if (volum > 150 && volum <= 175){
    pwm = 175;
}else if (volum > 175 && volum <= 200){
    pwm = 200;
}else if (volum > 200 && volum <= 225){
    pwm = 225;
}else if (volum > 225 && volum <= 250){
    pwm = 250;
}
return pwm;
}

```

```

int decodeIR() {
    int valorcanal = -1;
    switch(codi){
        case 1:
            valorcanal = 10; //més volum
            break;
        case 9:
            valorcanal = 12; //menys volum
            break;
        case 8:
            valorcanal = 13; //canal menys
            break;
        case 10:
            valorcanal = 11; //canal més
            break;
        case 12:
            valorcanal = 0; //canal 0
            break;
        case 16:
            valorcanal = 1; //canal 1
            break;
        case 17:
            valorcanal = 2; //canal 2
            break;
        case 18:
            valorcanal = 3; //canal 3
            break;
        case 20:
            valorcanal = 4; //canal 4
            break;
        case 21:
            valorcanal = 5; //canal 5
            break;
        case 22:
            valorcanal = 6; //canal 6
            break;
        case 24:
            valorcanal = 7; //canal 7

```

```

        break;
    case 25:
        valorcanal = 8; //canal 8
        break;
    case 26:
        valorcanal = 9; //canal 9
        break;
    }
    return valorcanal;
}

int volumIR() {
    int valorcanal = 0;
    switch(codi){
        case 1:
            valorcanal = 1;
            break;
        case 9:
            valorcanal = 0;
            break;
    }
    return valorcanal;
}

int mapCodeToButton(unsigned long code) {
    // Check for codes from this specific remote
    if ((code & 0x0000FFFF) == 0x0000BFF) {
        // No longer need the lower 16 bits. Shift the code by 16
        // to make the rest easier.
        code >>= 16;
        // Check that the value and inverse bytes are complementary.
        if (((code >> 8) ^ (code & 0x00FF)) == 0x00FF) {
            return code & 0xFF;
        }
    }
    return -1;
}

int readInfrared() {
    int result = -1;
    // Check if we've received a new code
    if (IrReceiver.decode()) {
        // Get the infrared code
        unsigned long code = IrReceiver.decodedIRData.decodedRawData;
        // Map it to a specific button on the remote
        result = mapCodeToButton(code);
        // Enable receiving of the next value
        IrReceiver.resume();
    }
    return result;
}

```

Explicació

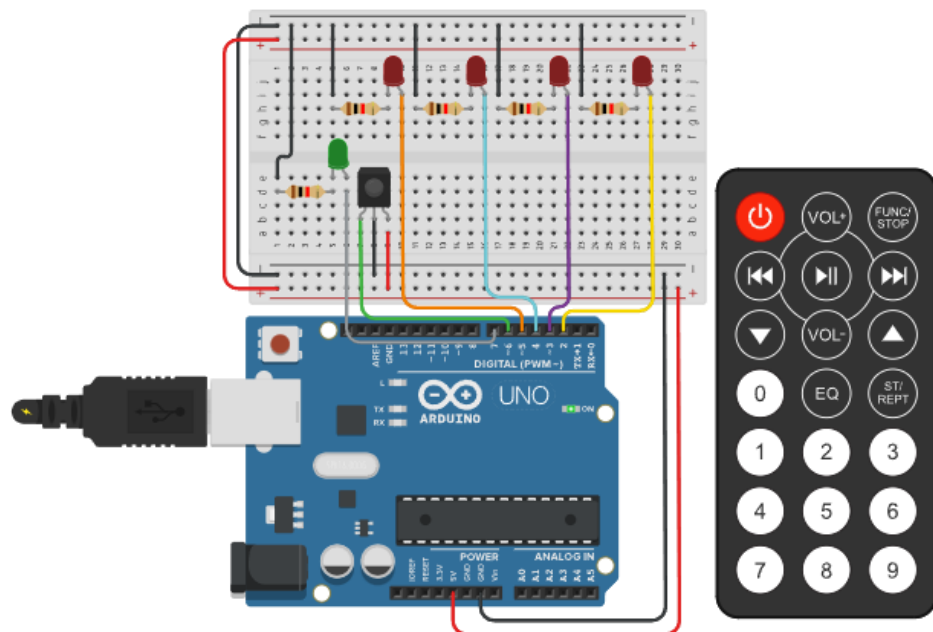
Per aquest apartat hem mantingut tot el codi que ja teníem en l'exercici anterior, però hem afegit dos elements nous.

El primer element, ja que volem poder interactuar a partir del canal sèrie, és una condició `if(Serial.available())` que s'encarrega de revisar contínuament si s'ingressa algun element. En cas afirmatiu el recull en la variable **received**.

El segon element que hem afegit, és la funció `serialtoinf()`. Aquesta funció té l'objectiu de passar l'element afegit per l'usuari al "llenguatge" del infraroig, amb el qual el nostre programa anterior ja sap com interactuar.

Una vegada l'usuari afegeix una acció, la passem al codi de l'infraroig i més endavant al programa per a que apliqui l'acció.

Muntatge



Link

<https://www.tinkercad.com/things/3oowPwFk7oj-ex1prac4/editel?returnTo=%2Fthings%2F3oowPwFk7oj-ex1prac4&sharecode=RrLijLJdqyuk5hFlkMNkIghaDh1dNGkPJ6pxZpvd1JI>