

```
const int clk = 2;
int estadoCLK = LOW;
long tiempoPasadoClk = 500;
long duty = 500;

const int tx = 3;
int estadoTx = LOW;
long tiempoPasadoTx = 500;
long intervaloTx = 500;

long tiempoPasadoHamming = 0;
long intervaloHamming = 0;

long tiempoPasadoLectura = 0;
long intervaloLectura = 0;

int estadoAnterior1 = LOW;

int estadoAnterior2 = LOW;

int estadoAnterior3 = LOW;

int estadoAnterior4 = LOW;

int dato[] = {0,0,0,0};
```

```
int hamtx[7] = {0};
```

```
int i = 0;
```

```
int l = 0;
```

```
int index = 0;
```

```
void setup() {  
    Serial.begin(9600);  
    pinMode(clk, OUTPUT);  
    pinMode(tx, OUTPUT);  
}
```

```
void loop() {  
    lectura();  
    transmisor(hamtx, sizeof(hamtx) /  
sizeof(int));  
}
```

```
void lectura() {  
    unsigned long tiempoActual = millis();  
    if(tiempoActual - tiempoPasadoLectura >
```

```
intervaloLectura) {  
    tiempoPasadoLectura = tiempoActual;  
  
    char receivedData[5];  
  
    bool validInput = true;  
  
    while (index < 4 && validInput) {  
        if (Serial.available() > 0) {  
            char receivedChar = Serial.read();  
  
            if (receivedChar == '0' ||  
receivedChar == '1') {  
                receivedData[index] =  
receivedChar;  
                index++;  
            }  
            else {  
                validInput = false;  
            }  
        }  
    }  
  
    if (!validInput || Serial.available() >
```

```
0) {  
    while (Serial.available() > 0) {  
        Serial.read();  
    }  
}
```

```
if (index == 4) {  
    receivedData[4] = '\0';  
    //Serial.println(receivedData);  
}
```

```
dato[0]=receivedData[0]- '0';  
dato[1]=receivedData[1]- '0';  
dato[2]=receivedData[2]- '0';  
dato[3]=receivedData[3]- '0';
```

```
if ((dato[0] != estadoAnterior1) ||  
(dato[1] != estadoAnterior2) || (dato[2] !=  
estadoAnterior3) || (dato[3] !=  
estadoAnterior4)) {  
    hamming();  
}
```

```
estadoAnterior1 = dato[0];
```

```

        estadoAnterior2 = dato[1];
        estadoAnterior3 = dato[2];
        estadoAnterior4 = dato[3];
    }
}
}
}
}

```

```

void hamming() {
    unsigned long tiempoActual = millis();
    if(tiempoActual - tiempoPasadoHamming >
intervaloHamming) {
        tiempoPasadoHamming = tiempoActual;

        Serial.println((String) "datos ->" +
dato[0] + dato[1] + dato[2] + dato[3] +
"\n");

        hamtx[0]=(dato[0]^dato[1])^dato[3];
        hamtx[1]=(dato[0]^dato[2])^dato[3];
        hamtx[2]=dato[0];
        hamtx[3]=(dato[1]^dato[2])^dato[3];
        hamtx[4]=dato[1];
        hamtx[5]=dato[2];
    }
}

```

```

    hamtx[6]=dato[3];

    Serial.println((String) "hamming(7,
4)\n\n" + "Bit/P1  " + "Bit/P2  " + "Bit3  "
+ "Bit/P4  " + "Bit5  " + "Bit6  " +
"Bit7");
    for (int i = 0; i < 7; i++) {
        Serial.print((String) "  " +
hamtx[i] + "  ");
    }
    Serial.println("\n");

    //
}

}

```

```

void transmisor(int bits[], int longitud) {
    unsigned long tiempoActual = millis();
    if(tiempoActual - tiempoPasadoTx >
intervaloTx) {
        tiempoPasadoTx = tiempoActual;

        for (l; l < longitud; l++) {

```

```

Reloj();
if(estadoCLK==HIGH){
    digitalWrite(tx, bits[l]);
    Serial.print(bits[l++]);
/*    if(l==2){
        bits[2]=0;
    }*/
}
else{
    digitalWrite(tx, 0);
}
if(l>6){
    Serial.print(" -> Hamming Enviado
\n\n
\n");
    l=0;
    i=0;
    index = 0;

}
break;
}
}

```

```
}  
  
void Reloj() {  
    unsigned long tiempoActual = millis();  
    if(tiempoActual - tiempoPasadoClk > duty)  
    {  
        tiempoPasadoClk = tiempoActual;  
        if (estadoCLK == LOW) {  
            estadoCLK = HIGH;  
        }  
        else {  
            estadoCLK = LOW;  
        }  
        digitalWrite(clk, estadoCLK);  
    }  
}
```