

```
const int clk = 2;
int estadoCLK = LOW;
long tiempoPasadoClk = 0;
long duty = 0;

const int tx = 33;
int estadoTx = LOW;
long tiempoPasadoTx = 0;
long intervaloTx = 0;

long tiempoPasadoHamming = 0;
long intervaloHamming = 0;

long tiempoPasadoLectura = 0;
long intervaloLectura = 0;

int sw4=53;
int sw3=51;
int sw2=49;
int sw1=47;

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;
```

```
void setup() {
```

```
    Serial.begin(9600);
```

```
    pinMode(clk, OUTPUT);
```

```
    pinMode(tx, OUTPUT);
```

```
    pinMode(sw1, INPUT);
```

```
    pinMode(sw2, INPUT);
```

```
    pinMode(sw3, INPUT);
```

```
    pinMode(sw4, INPUT);
```

```
}
```

```
void loop() {
```

```
    lectura();
```

```
    transmissor(hamtx, sizeof(hamtx) /
sizeof(int));

}

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

        dato[0]=digitalRead(sw1);
        dato[1]=digitalRead(sw2);
        dato[2]=digitalRead(sw3);
        dato[3]=digitalRead(sw4);

        if ((dato[0] != estadoAnterior1) ||
(dato[1] != estadoAnterior2) || (dato[2] !=
estadoAnterior3) || (dato[3] !=
estadoAnterior4)) {
            i=0;
            l=0;
            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++]);
    }
    else{
        digitalWrite(tx, 0);
    }
    if(l>6) {
        Serial.print(" -> Hamming Enviado
\n\n
\n");
    }
    break;
}
}
}
}

```

```

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

```

```
        estadoCLK = LOW;
    }
    digitalWrite (clk, estadoCLK);
}
}
```