```
#define dataPin 3
#define clockPin 2
byte hamrx[7]={0};
byte bitCount = 0;
int data[3] = {0};
int error = 0;
int pE=0;
int p[3] = {0};

void invhamming() {

  Serial.print("HAMMING RX: ");
  for (byte i = 0; i < 7; i++) {
    Serial.print(hamrx[i]);
  }
  Serial.println();

  p[0] = hamrx[2]^hamrx[4]^hamrx[6];
  p[1] = hamrx[2]^hamrx[5]^hamrx[6];
  p[2] = hamrx[4]^hamrx[5]^hamrx[6];

  if (p[0]!=hamrx[0]){
    pE = pE+1;
  }
  if (p[1]!=hamrx[1]){
    pE = pE+2;
  }
  if (p[2]!=hamrx[3]){
    pE = pE+4;
  }
  if (pE!=0){
    hamrx[pE-1]= !hamrx[pE-1];
    Serial.println((String)"ERROR
DETECTADO EN EL BIT:" + (pE));
    error = 1;
    pE=0;
  }

  data[0] = hamrx[2];
  data[1] = hamrx[4];
  data[2] = hamrx[5];
  data[3] = hamrx[6];
}

void print_vector() {
  Serial.println((String)"ERRORES
DETECTADOS: "+error);
  error = 0;
  Serial.print("HAMMING CORRECION: ");
  for (byte i = 0; i < 7; i++) {
    Serial.print(hamrx[i]);
  }
  Serial.print("\nDATO: ");
  for (byte i = 0; i < 4; i++) {
    Serial.print(data[i]);
  }
  Serial.println();
}

void clock_reciver() {
  byte stateClock = digitalRead(clockPin);
  if (stateClock == HIGH) {
    hamrx[bitCount] = digitalRead(dataPin);
    bitCount++;
  }
  if (bitCount == 7) {
    invhamming();
    print_vector();
    bitCount = 0;

  }
}
void setup() {
  pinMode(dataPin, INPUT);
  pinMode(clockPin, INPUT);
  Serial.begin(9600);

attachInterrupt(digitalPinToInterrupt(clock
Pin), clock_reciver, RISING);
}
void loop() {
}
```