

# Análisis de Señales

## Curso 2020

Aplicación Teorema del muestreo y CAD

Instrumentación

Prof. Jorge Runco

# Introducción

Temperatura,  
presión, energía



Magnitud física a medir

Transductor,  
sensor



Genera tensión o corriente  
función de la magnitud física  
a medir

Acondicionar la  
señal



Amplificación, filtrado,  
conversión A/D

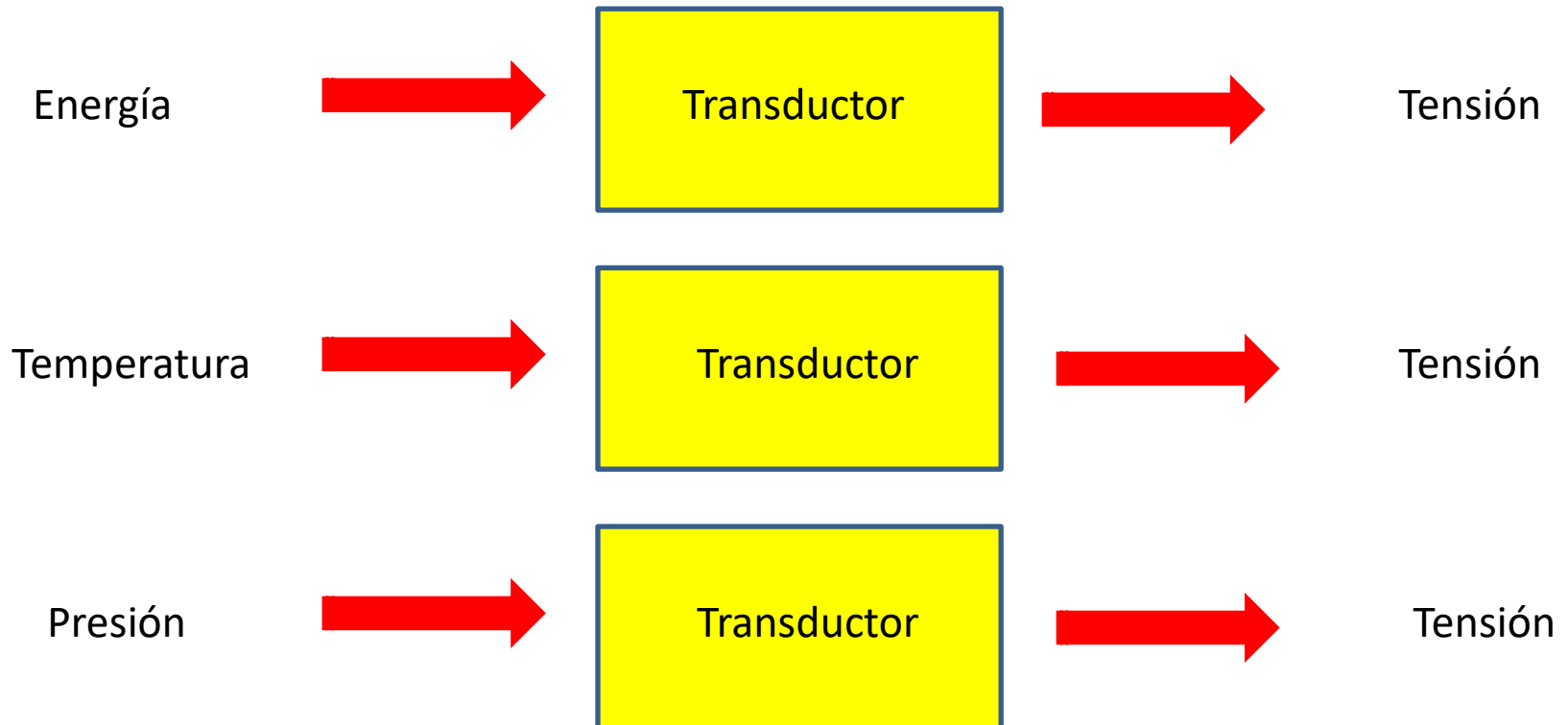
Transmisión



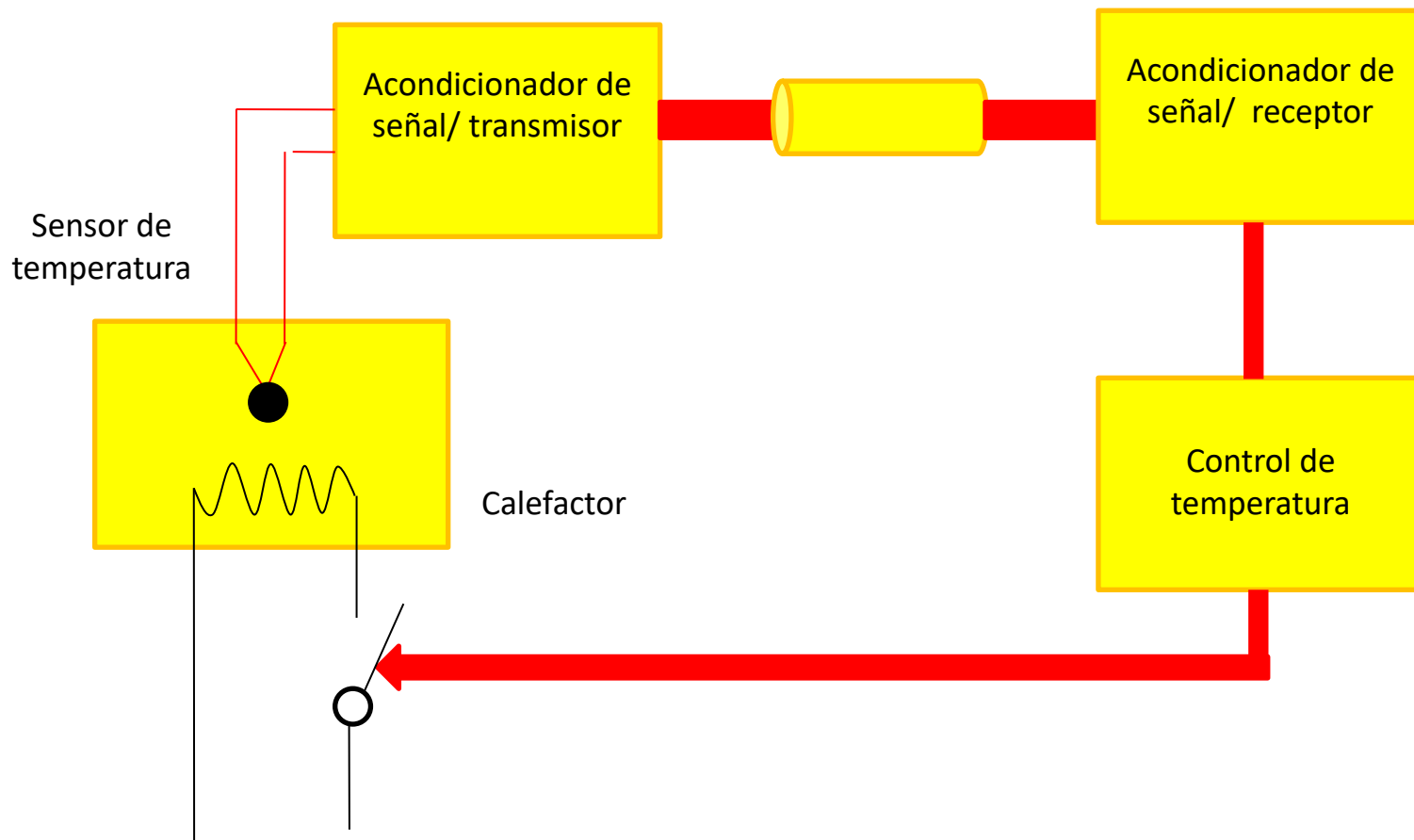
Medio de transmisión

# Transductor

- Dispositivo que convierte una señal física de entrada en una señal de salida de tipo: eléctrico (tensión, corriente).



# Esquema en bloques de un control de temperatura



- La primera tarea a realizar por el acondicionador de señal, será amplificar las débiles señales entregadas por el transductor, hasta niveles utilizables por el resto de la cadena de medida.
- El amplificador de señales débiles que realice esta tarea será, por lo tanto, un componente crítico (y fundamental) del sistema de instrumentación.

- Hoy en día la mayoría de los amplificadores para sistemas de instrumentación se realizan utilizando amplificadores operacionales, por su versatilidad y sencillez de utilización.

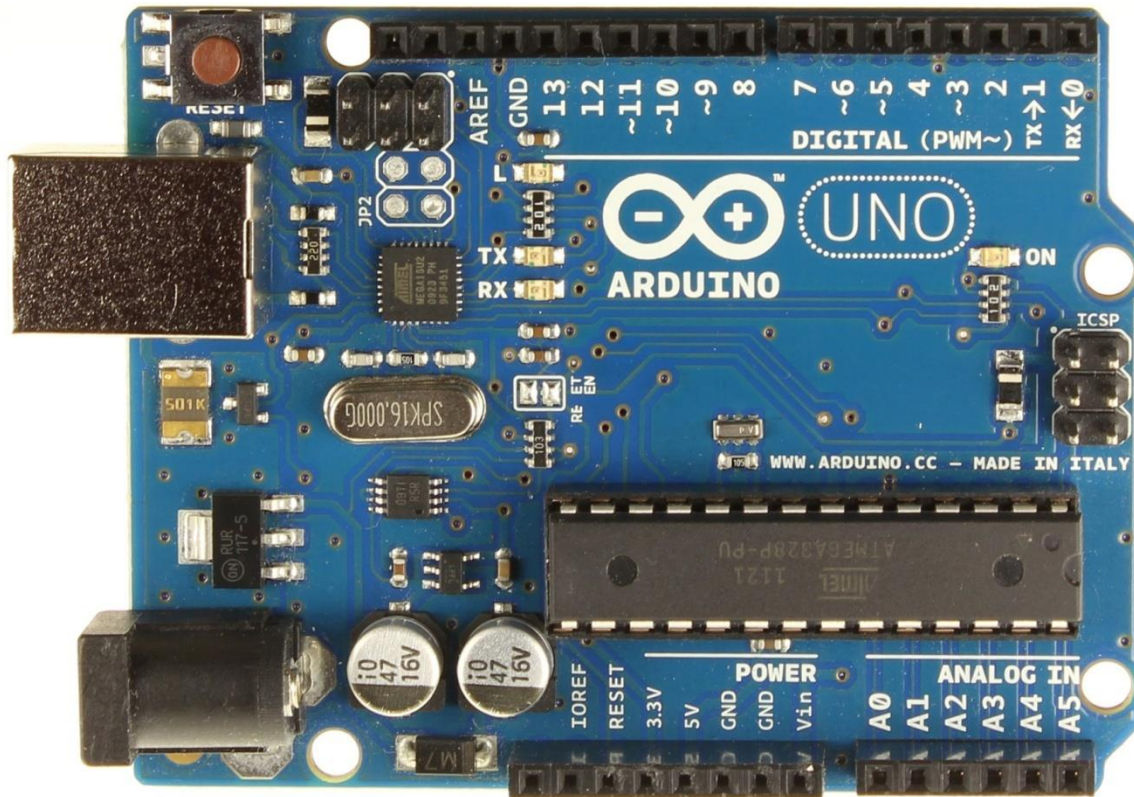
# Transductor

- Es un dispositivo que convierte la magnitud física a medir en una señal eléctrica como una tensión o una corriente.
- Ejemplos
  - Termocupla tipo K  $40 \mu\text{V}/^{\circ}\text{C}$
  - LM35  $10 \text{ mV}/^{\circ}\text{C}$

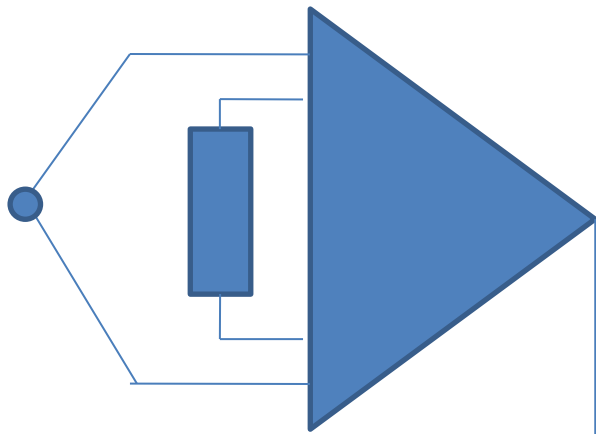
- El transductor entrega una tensión que es función de la magnitud física a medir.
- Para almacenar esta magnitud física en el sistema de computo debemos convertirla en un número (bits), usamos un conversor analógico digital.
- Luego hacemos una cuenta y de acuerdo a alguna condición, tomamos una decisión.



# Para aplicaciones dedicadas:

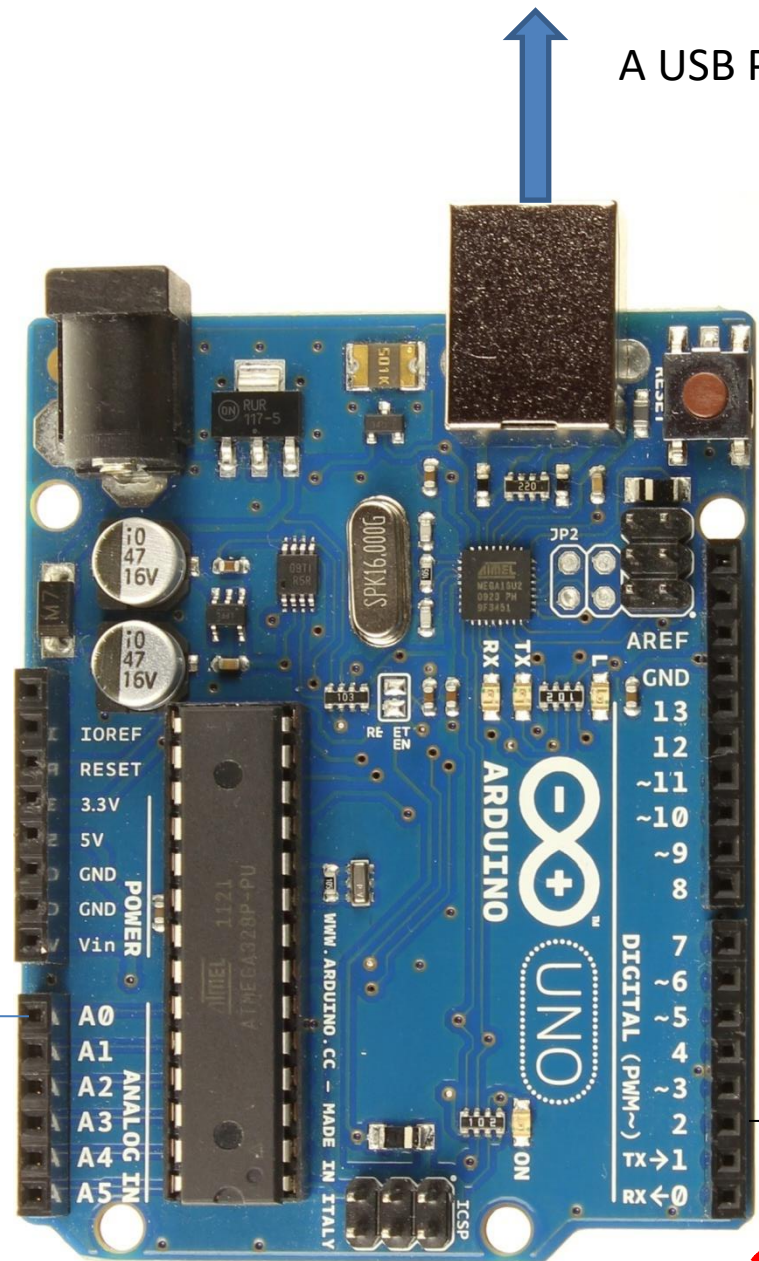


Sensor de temperatura

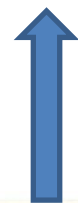


Amplificador de instrumentación

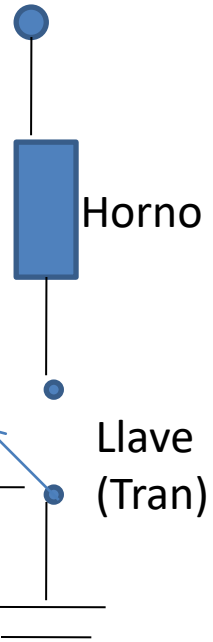
Entrada analógica



A USB PC



Fuente de alimentación



Horno

Llave (Tran)

Salida digital



# ¿Cómo elegir un conversor AD?

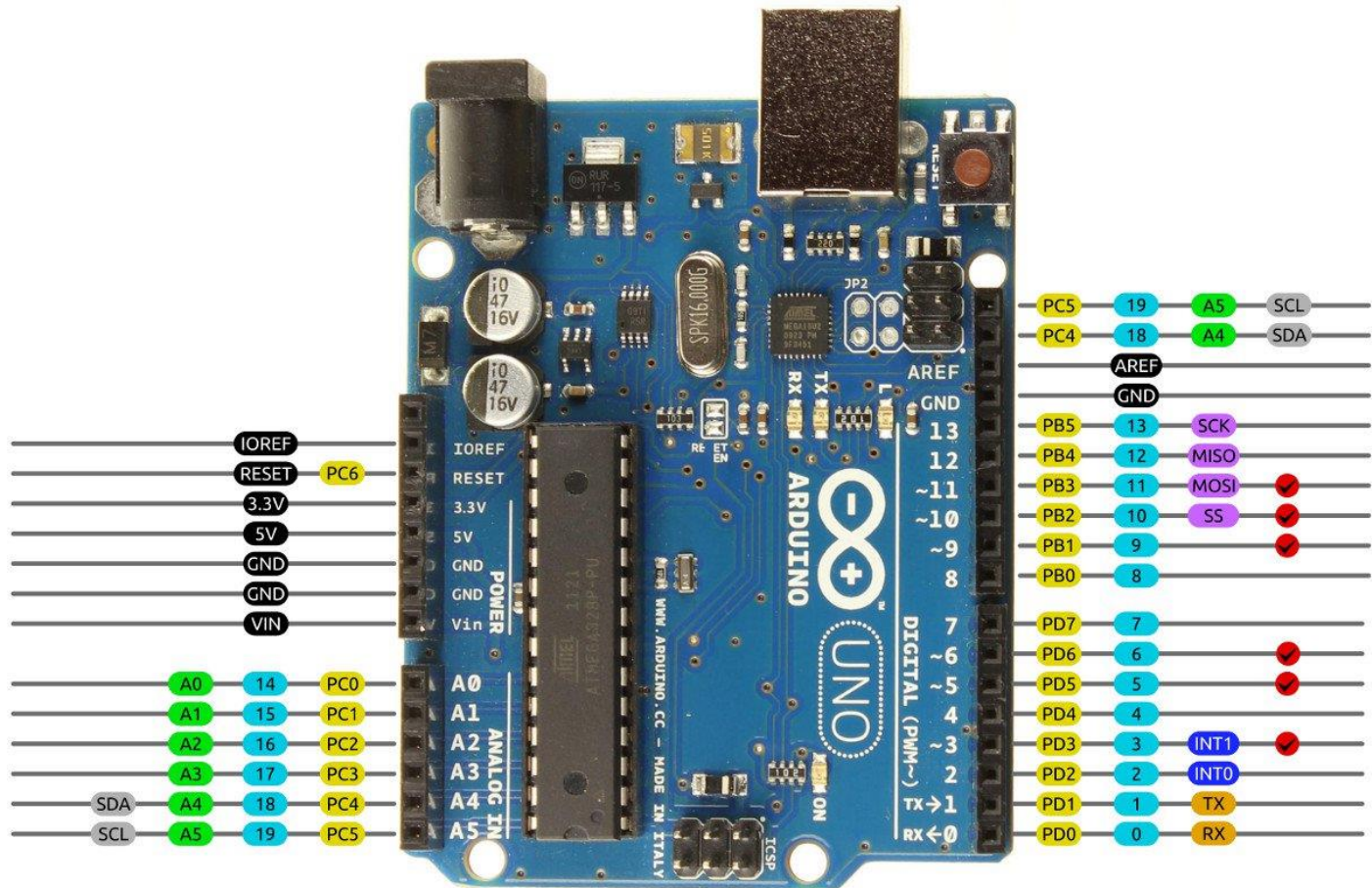
- $0^{\circ} \text{ C} \rightarrow 0 \text{ volt}$
- $500^{\circ} \text{ C} \rightarrow 5 \text{ volt}$  (máx tensión en el conversor)
- Error =  $0.5^{\circ} \text{ C}$
- CAD  $\rightarrow$  8 bits, 9 bits y 10 bits
- 8 bits  $\rightarrow 500/2^8 = 500/256$  aprox  $2^{\circ} \text{ C}$
- 9 bits  $\rightarrow 500/2^9 = 500/512$  aprox  $1^{\circ} \text{ C}$
- 10 bits  $\rightarrow 500/2^{10} = 500/1024$  aprox  $0.5^{\circ} \text{ C}$

# ¿Cómo elegir un conversor AD?

- Tiempo de conversión =  $T_c$
- $T_c = 1 \text{ mseg} \rightarrow f_{\text{muestreo}} = 1/1 = 1000 \text{ Hz}$
- $f_{\text{muestreo}} = 2 f_{\text{max}} \rightarrow f_{\text{max}} = 500 \text{ Hz}$
- $T_c = 1 \text{ useg} \rightarrow f_{\text{muestreo}} = 1/1 = 1 \text{ MHz}$
- $f_{\text{muestreo}} = 2 f_{\text{max}} \rightarrow f_{\text{max}} = 500 \text{ kHz} = 0.5 \text{ MHz}$



# Arduino Uno R3 Pinout



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT



2014 by Bouni  
Photo by Arduino.cc

```
int T0=0;
int P1=0;
int opcion=0;
String medidas;
int ValvulaPV1=2;

void setup() {
  //Inicializa puerto serie
  Serial.begin(115200);
  pinMode(ValvulaPV1, OUTPUT);
}
void loop() {
  if (Serial.available()>0)
  {
    opcion=Serial.read();

    switch (opcion)
    {
```

```
      case ('1'):
        T0=analogRead(A1);
        medidas+=T0;
        medidas+=":";
        delay(1);
        P1=analogRead(A2);
        medidas+=P1;
        medidas+=":";
        delay(1);
        Serial.println(medidas);
        delay(200);
        medidas="";
        break;
      case('2'):
        digitalWrite(ValvulaPV1,HIGH);
        break;
      case('A'):
        digitalWrite(ValvulaPV1,LOW);
        break;
    }
  }
}
```

```
python -m pip install PySerial
```

```
void setup() {  
  Serial.begin(9600);  
}  
void loop() {  
  Serial.println("Hola mundo");  
  delay(1000);  
}
```

```
import serial, time  
arduino = serial.Serial('COM4', 9600)  
time.sleep(2)  
rawString = arduino.readline()  
print(rawString)  
arduino.close()
```

```
import serial, time  
arduino = serial.Serial("COM4", 9600)  
time.sleep(2)  
arduino.write(b'9')  
arduino.close()
```



Type of Output Data	Output Data Format
Non-reading queries	<80 ASCII character string
Single reading (IEEE-488)	SD.DDDDDDDDESDD<nl>
Multiple readings (IEEE-488)	SD.DDDDDDDDESDD,..... <nl>
Single reading (RS-232)	SD.DDDDDDDDESDD<cr><nl>
Multiple readings (RS-232)	SD.DDDDDDDDESDD,.....<cr><nl>
	<b>S</b> Negative sign or positive sign
	<b>D</b> Numeric digits
	<b>E</b> Exponent
	<nl> newline character
	<cr> carriage return character

```
lectura1:=(lectura1+(devuelto[r]-$30)*Power(10,E));
```

```
r:=r-1;
```

```
E:=E+1;
```

```
for k:=1 to 17 do
```

```
devuelto1:=devuelto1+Chr(devuelto[k]);
```

```
    t2h:=devuelto[2];
```

```
    t2h:=t2h-48;
```

```
    t4h:=devuelto[4];
```

```
    t4h:=t4h-48;
```

```
    t5h:=devuelto[5];
```

```
    t5h:=t5h-48;
```

```
    t15h:=devuelto[15];
```

```
    t15h:=t15h-48;
```

# Otro instrumento.....

```
fa:=devueltof[12];
    if fa <= 57 then fa:=fa-48
        else fa:=fa-55;
fb:=devueltof[13];
    if fb <= 57 then fb:=fb-48
        else fb:=fb-55;
fc:=devueltof[14];
    if fc <= 57 then fc:=fc-48
        else fc:=fc-55;

fd:=devueltof[15];
    if fd <= 57 then fd:=fd-48
        else fd:=fd-55;

lecflujo:=fd+fc*16+fb*256+fa*4096;

lecflujo1:=(lecflujo/32000)*100;
```