

```

LDI R16, 0b0000_0110      ;Los bits en 1 son las salidas
                             ; (LED)

OUT DDRD,R16

LEYENDO_BOTONES:
IN R17, PIND               ;Esta línea lee el voltaje
                             ;presente en el puerto

LDI R18, 0b0000_1001      ;En R18 nos interesa leer
                             ;solamente PD0 y PD3 donde
AND R17, R18               ;están los push-buttons

```

La instrucción AND hará el siguiente proceso: suponiendo que el registro de lectura R17 posee este dato 0b0000_0111, es decir ya se había pulsados el push-button-1 y se prendió el LED-1, entonces:

```

AND 0b0000_0111 ← R18
    0b0000_1001 ← R17
    -----
    0b0000_0001

```

La instrucción AND coloca en 1 cuando ambos datos son 1, y coloca en 0 cuando ambos datos son desiguales

```

AND 0b0000_1100 ← R17
    0b0000_1001 ← R18
    -----
    0b0000_1000

```

```

AND 0b0000_1111 ← R18
    0b0000_1001 ← R17
    -----
    0b0000_1001

```

Presionando la tecla F1 del teclado, en el AVR Studio una ventana de ayuda emergerá, y se podrá ver ahí la sintaxis y lo referente a la instrucción **AND**, dónde se almacena el resultado, etc.

**;ESTE PROGRAMA HABILITARÁ UN SOLO PUERTO PARA ENTRADA Y SALIDA
;CONECTANDO AL "PUERTO D": 2 PUSH-BUTTONS Y 2 LED, USANDO LA TÉCNICA
;DEL ENMASCARAMIENTO (MÁSCARA PARA SÓLO ENTRADAS).**

```

.INCLUDE "TN2313DEF.INC"
.CSEG
.ORG 0

```

```

LDI R16, LOW(RAMEND)
OUT SPL, R16

```

Stack Pointer

```

LDI R16, 0b0000_0110
OUT DDRD, R16

```

PD1 y PD2 son salidas, y los demás bits son entradas

LEYENDO:

```

IN R17, PIND

```

R17 contiene el valor leído del "puerto D"

```

LDI R18, 0b0000_1001
AND R17, R18

```

Hacemos la máscara sólo para los bits de entrada que serán para PD0 y PD3, y procedemos con la instrucción **AND** entre el valor leído de R17 con la máscara de R18. El resultado está en R17

```

LDI R19, 0b0000_0001
CP R17, R19
BR EQ PREDER_LED1

```

Cargamos a R19 con el valor del push-button de encendido del LED-1, y comparamos con R17

```
LDI R19,0b0000_1000
CP R17,R19
BREQ PREDER_LED2
```

Cargamos a R19 con el valor del push-button de encendido del LED-2, y comparamos con R17

```
RJMP LEYENDO
```

```
PREDER_LED1:
LDI R20,0b0000_0010
OUT PORTD,R20
RJMP LEYENDO
```

```
PREDER_LED2:
LDI R20,0b0000_0100
OUT PORTD,R20
RJMP LEYENDO
```