

Encabezado para ATmega8515

Stack Pointer para ATmega8515

```
LDI R16,0b0000_0011          ;PB0=UP/DOWN, PB1=INCREMENTO
OUT DDRB,R16
```

```
;INICIALIZAR POTENCIÓMETRO DIGITAL
```

```
LDI R17,100
```

```
LDI R18,0
```

```
INICIALIZAR:
```

```
;0b0000_00 1 1
;          INC U/D
```

```
LDI R16,0b0000_0000
OUT PORTB,R16
```

```
RCALL DELAY_2_MICROS
```

```
LDI R16,0b0000_0010
OUT PORTB,R16
```

```
RCALL DELAY_2_MICROS
```

```
INC R18
CP R17,R18
BRQ SUBIDA
RJMP INICIALIZAR
```

```
RCALL DELAY_MULTIMETRO
;0b0000_00 1 1
; INC U/D
```

```
SUBIDA:
LDI R17,100
LDI R18,0
```

```
CONTINUA_1:
```

```
LDI R16,0b0000_0001
OUT PORTB,R16
```

```
RCALL DELAY_2_MICROS
```

```
LDI R16,0b0000_0011
OUT PORTB,R16
```

```
RCALL DELAY_2_MICROS
```

```
RCALL DELAY_MULTIMETRO
```

```
INC R18
CP R17,R18
BRQ BAJADA
RJMP CONTINUA_1
```

```
BAJADA:
LDI R17,100
LDI R18,0
```

CONTINUA\_2:

```
;0b0000_00 1 1
;      INC U/D
LDI R16,0b0000_0000
OUT PORTB,R16
```

RCALL DELAY\_2\_MICROS

```
LDI R16,0b0000_0010
OUT PORTB,R16
```

RCALL DELAY\_2\_MICROS

RCALL DELAY\_MULTIMETRO

```
INC R18
CP R17,R18
BREQ SUBIDA
RJMP CONTINUA_2
```

```
DELAY_2_MICROS:
NOP
RET
```

DELAY\_MULTIMETRO:

```
.
.
.
```

RET

### Programa:

```
; MIDE EL VOLTAJE ANALÓGICO DE UN CONVERTIDOR ADC0804
;Y MUESTRA EL VOLTAJE LEÍDO EN UN DISPLAY LCD
```

```
.INCLUDE "M8515def.inc"

.DEF DIVIDENDO          = R17 ;El número que proviene del
                           ;ADC0804
.DEF RESIDUO            = R18 ;RESIDUO
.DEF DIVISOR            = R19 ;=9
.DEF RESULTADO          = R20 ;RESULTADO
.DEF DECENAS            = R21 ;DECENAS DEL VOLTAJE
.DEF UNIDADES           = R22 ;UNIDADES DEL VOLTAJE

.CSEG
.ORG 0
```

#### Stack Pointer para ATmega8515

```
LDI R16,$00                      ;ENTRADA DEL CONVERTIDOR
                                   ;ADC0800
OUT DDRA,R16

LDI R16,0b0000_0011             ;CONTROL DE POT-DIGITAL
OUT DDRB,R16

LDI R16,$FF                      ;LCD
OUT DDRC,R16

LDI R16,$FF                      ;LCD
OUT DDRE,R16

INICIALIZANDO_POT_DIGITAL:
LDI R26,100
LDI R27,0
LDI R31,0                        ;ALMACENA EL INCREMENTO DEL POT

RESET:
LDI R16,0b0000_0010
OUT PORTB,R16
RCALL DELAY_POT
LDI R16,0b0000_0000
OUT PORTB,R16
RCALL DELAY_POT
INC R27
CP R27,R26
BRNE RESET

;INICIALIZANDO DISPLAY
```

```
RCALL DISPLAY_CONTROL_ON
RCALL CLEAR_DISPLAY
RCALL ESCRIBIR_EN_LINEA_UNO
RCALL HOME
```

```
RCALL C
RCALL O
RCALL L
RCALL O
RCALL C
RCALL A
RCALL ESPACIO
RCALL R
RCALL E
RCALL S
RCALL I
RCALL S
RCALL T
RCALL O
RCALL R
```

```
RCALL ESCRIBIR_EN_LINEA_DOS
```

```
RCALL P
RCALL R
RCALL E
RCALL S
RCALL I
RCALL O
RCALL N
RCALL A
RCALL ESPACIO
RCALL S
RCALL T
RCALL A
RCALL R
RCALL T
```

```
LEYENDO_START:                ;Se introduce un BOTÓN de
                                ;arranque de prueba
```

```
IN R0,PINB
LDI R16,0b0000_0100
AND R0,R16
CP R0,R16
BRNE LEYENDO_START
```

```

;*****
LEYENDO_RESISTENCIA:
IN R30, PINA ;R30-LECTURA DE CONVERTIDOR
LDI R16,0
CP R30,R16
BRQ ESCRIBIR_OHMS
RCALL INCREMENTANDO_POT
RCALL DELAY_POT
RJMP LEYENDO_RESISTENCIA

INCREMENTANDO_POT:
LDI R16,0b0000_0011
OUT PORTB,R16
RCALL DELAY_POT
LDI R16,0b0000_0001
OUT PORTB,R16
RCALL DELAY_POT

INC DIVIDENDO

RET

ESCRIBIR_OHMS:
;SUBROUTINA DE DIVISIÓN de Gerhard Schmidt
;Se cargan los registros correspondientes según
;la estructura de la división:

;          COCIENTE (RESULTADO)
;
;          _____
;  DIVISOR | DIVIDENDO
;          RESIDUO
;
;

LDI DIVISOR,9 ;DIVISOR

DIV8:
    CLR RESIDUO ;BORRA EL RESIDUO
    CLR RESULTADO ;BORRA EL RESULTADO
    INC RESULTADO ;INCREMENTA PARA EMPEZAR

DIV8A:
    CLC ;BORRA LA BANDERA "CARRY"
    ROL DIVIDENDO ;ROTA A LA IZQUIERDA CON EL
                  ;"CARRY"

```

```

    ROL RESIDUO                ;ROTA A LA IZQUIERDA CON EL
                                ;"CARRY"
    BRCS DIV8B                 ;UN "1" SE MOVIÓ A LA IZQUIERDA
    CP RESIDUO,DIVISOR         ;DIVISIÓN SI EL RESULTADO ES
                                ;0 ó 1
    BRCS DIV8C                 ;SALTA SI ES MENOR
DIV8B:
    SUB RESIDUO,DIVISOR        ;RESTA DEL DIVISOR Y RESIDUO
    SEC                        ;ACTIVA EL "CARRY"
    RJMP DIV8D
DIV8C:
    CLC                        ;BORRA LA BANDERA "CARRY"
DIV8D:
    ROL RESULTADO              ;ROTA A LA IZQUIERDA CON EL
                                ;"CARRY"
    BRCC DIV8A                 ;SALTAR A "DIV8A" SI "CARRY" ES
                                ;BORRADA

RCALL CLEAR_DISPLAY
RCALL HOME

RCALL R
RCALL E
RCALL S
RCALL U
RCALL L
RCALL T
RCALL A
RCALL D
RCALL O
RCALL ESPACIO
RCALL IGUAL

RCALL ESCRIBIR_EN_LINEA_DOS

LDI R16,$30
ADD RESULTADO,R16

OUT PORTC,RESULTADO           ;Sacando ENTEROS en LCD
LDI R16,$05
OUT PORTE,R16
RCALL DELAY
LDI R16,$00
OUT PORTE,R16

```

RCALL PUNTO

LDI R16,\$30  
ADD RESIDUO,R16

OUT PORTC,RESIDUO ;Sacando DECIMALES en LCD  
LDI R16,\$05  
OUT PORTE,R16  
RCALL DELAY  
LDI R16,\$00  
OUT PORTE,R16

RCALL K

LDI R16,\$F4 ;Sacando SÍMBOLO de OHMS  
OUT PORTC,R16  
LDI R16,\$05  
OUT PORTE,R16  
RCALL DELAY  
LDI R16,\$00  
OUT PORTE,R16

FIN: RJMP FIN

IGUAL:  
LDI R16,\$3D ;HEXADECIMAL PARA EL SÍMBOLO "IGUAL"  
OUT PORTC,R16

.  
.  
.

RET

.INCLUDE "CODIGOS\_ASCII\_Y\_CONTROL\_LCD.TXT"