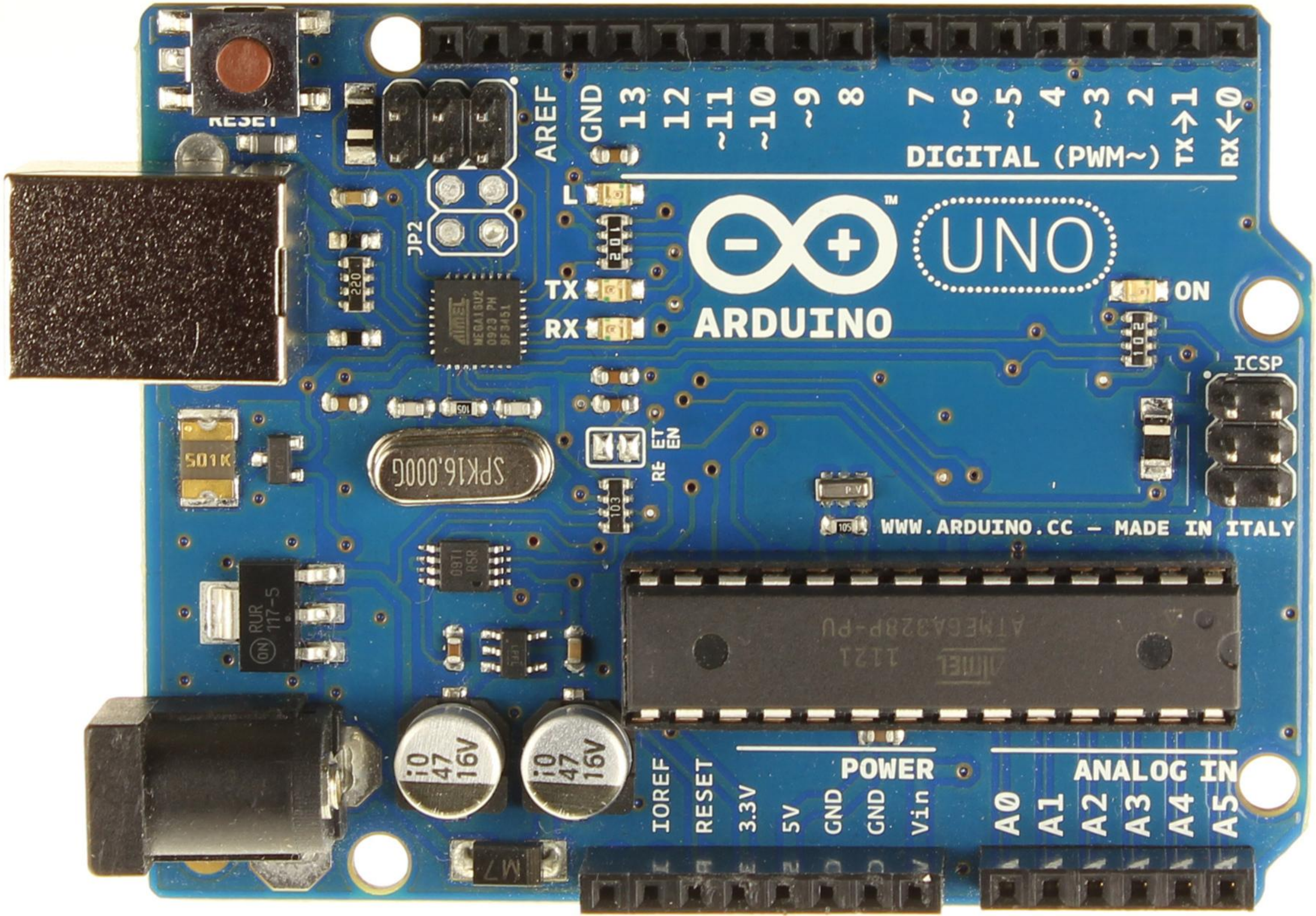
Curso avanzado sobre Arduino: Ensamblador

ElCacharreo.com

ElCacharreo.com

Arduino Intermedio: Presente



Arduino Intermedio: Presente

José Antonio Vacas Martínez

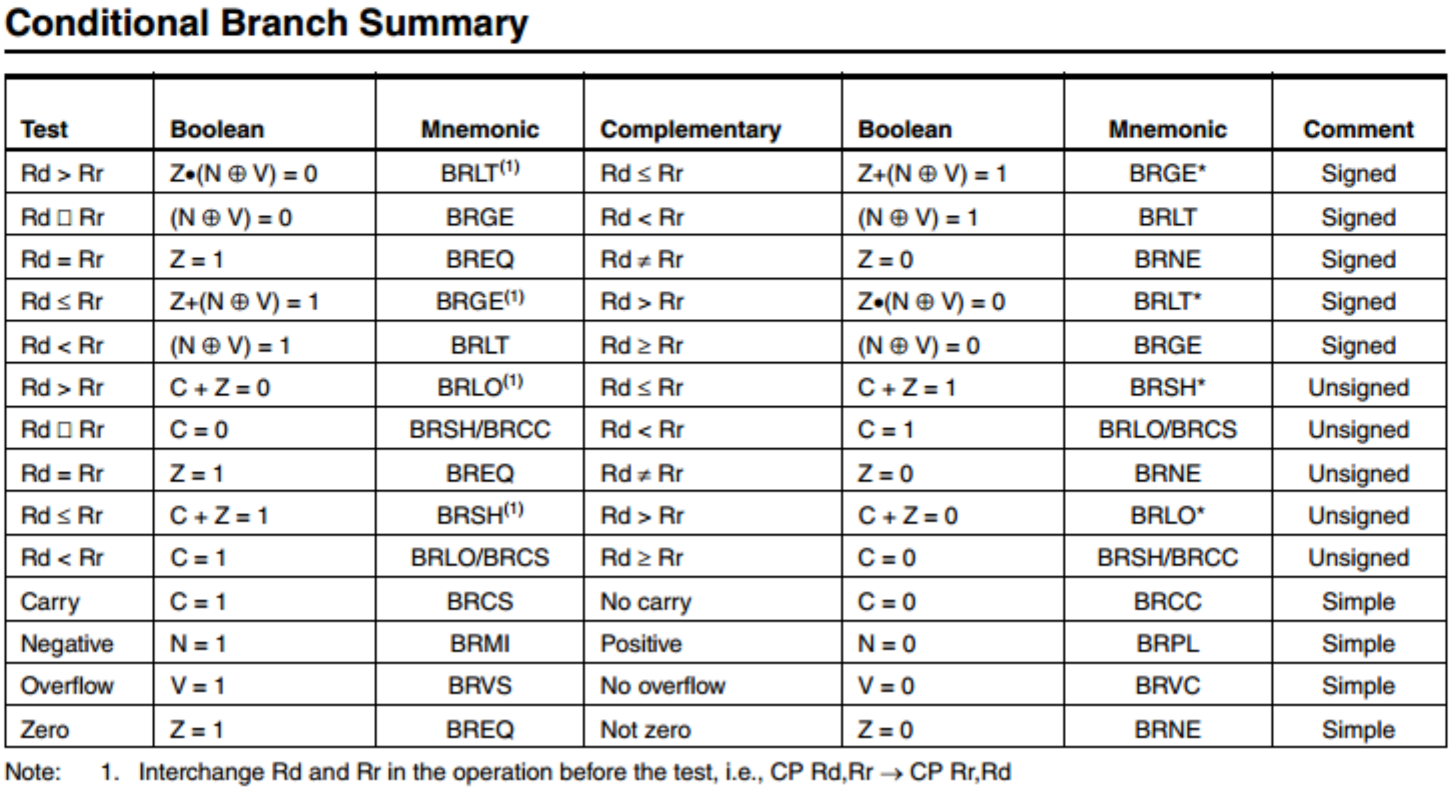
blog

javacasm@elcacharreo.com

twitter

linkedin

Arduino Avanzado: Assembler



Conditional Branch Summary

Test	Boolean	Mnemonic	Complementary	Boolean	Mnemonic	Comment
Rd > Rr	Z•(N ⊕ V) = 0	BRLT[1]	Rd ≤ Rr	Z+(N ⊕ V) = 1	BRGE*	Signed
Rd □ Rr	(N ⊕ V) = 0	BRGE	Rd < Rr	(N ⊕ V) = 1	BRLT	Signed
Rd = Rr	Z = 1	BREQ	Rd ≠ Rr	Z = 0	BRNE	Signed
Rd ≤ Rr	Z+(N ⊕ V) = 1	BRGE[1]	Rd > Rr	Z•(N ⊕ V) = 0	BRLT*	Signed
Rd < Rr	(N ⊕ V) = 1	BRLT	Rd ≥ Rr	(N ⊕ V) = 0	BRGE	Signed
Rd > Rr	C + Z = 0	BRLO[1]	Rd ≤ Rr	C + Z = 1	BRSH*	Unsigned
Rd □ Rr	C = 0	BRSH/BRCC	Rd < Rr	C = 1	BRLO/BRCS	Unsigned
Rd = Rr	Z = 1	BREQ	Rd ≠ Rr	Z = 0	BRNE	Unsigned
Rd ≤ Rr	C + Z = 1	BRSH[1]	Rd > Rr	C + Z = 0	BRLO*	Unsigned
Rd < Rr	C = 1	BRLO/BRCS	Rd ≥ Rr	C = 0	BRSH/BRCC	Unsigned
Carry	C = 1	BRCS	No carry	C = 0	BRCC	Simple
Negative	N = 1	BRMI	Positive	N = 0	BRPL	Simple
Overflow	V = 1	BRVS	No overflow	V = 0	BRVC	Simple
Zero	Z = 1	BREQ	Not zero	Z = 0	BRNE	Simple

Note: 1. Interchange Rd and Rr in the operation before the test, i.e., CP Rd,Rr → CP Rr,Rd

Arduino Avanzado: Assembler

¿Realmente vale la pena?

Arduino Avanzado: Assembler

```
/* asmtest.h
 * Global register variables.
 */
#ifdef __ASSEMBLER__

/* Assembler-only */

#else /* !AS

/* C-only

#include

extern "C" uint8_t led(uint8_t);
extern "C" uint8_t asminit(uint8_t);

#endif /* ASSEMBLER */
```

```
#include "asmtest.h"
void setup()
{ asminit(0);}
void loop()
{ led(0);
delay(1000);
led(1);
delay(1000);}
```

```
#include "avr/io
#include "asmtest
; Define the function asmin
.global asm
asminit:
sbi 4,5; 4 = DDRB (0x24 - 0x20). Bit 5 = pin
ret
; Define the function le
.global led ; The assembly function must be declared
global
led:
cpi r24, 0x00 ; Parameter passed by caller in
breq turr
sbi 5, 5; 5 = PORTB (0x25 - 0x20). Bit 5 = pin
ret
turnoff:
cbi 5, 5; 5 = PORTB (0x25 - 0x20). Bit 5 = pin
ret
```

http://rwf.co/dokuwiki/doku.php?id=smallcpus

Arduino Avanzado: Assembler

SBI = Set Bit Immediate
Usage: address-0x20

CBI = Clear Bit Immediate
Usage: address-0x20

Enlaces

http://forum.arduino.cc/index.php/topic,40916.0.html

blink http://forum.arduino.cc/index.php/topic,4114.0.html

http://forum.arduino.cc/index.php/topic,4835.0.html

Conclusiones

Gracias por vuestra atención