

29-3-2017

Practica 6

Programación en lenguaje
Ensamblador del ATmega2560



CARLOS OMAR CALDERON MEZA

MICROPROCESADORES Y MICROCONTROLADORES
INGENIERO EN COMPUTACION – FACULTAD DE CIENCIAS
QUIMICAS E INGENIERIA

Objetivo:

Mediante esta práctica el alumno aprenderá el uso básico de programa AVR Studio como herramientas de programación en lenguaje Ensamblador para el sistema ATmega1280.

Equipo:

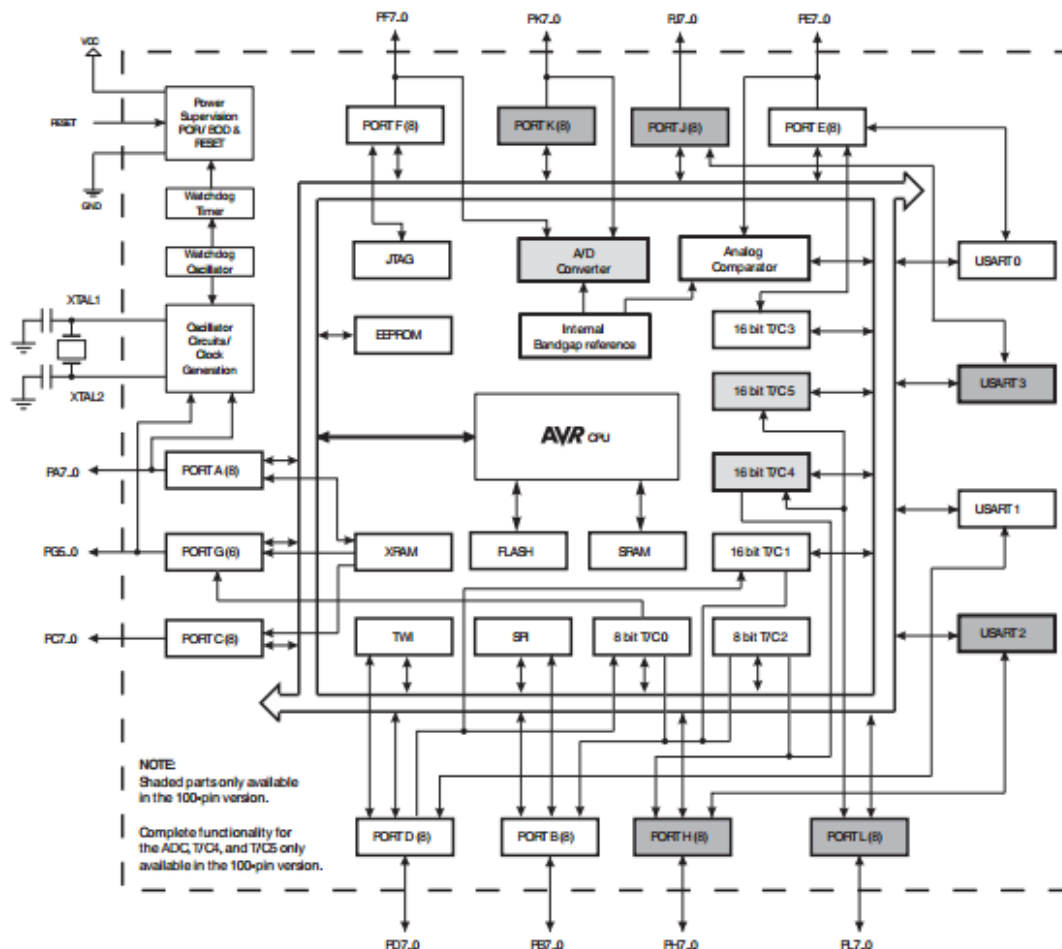
- Computadora Personal

Teoría:

- Arquitectura interna y conjunto de instrucciones del ATmega1280

Los microcontroladores de la familia ATmega640/1280/1281/2560/2561 son microcontroladores CMOS de 8 bits, todos estos modelos son de arquitectura RICS.

Diagrama a bloques del microcontrolador ATmega640/1280/1281/2560/2561:

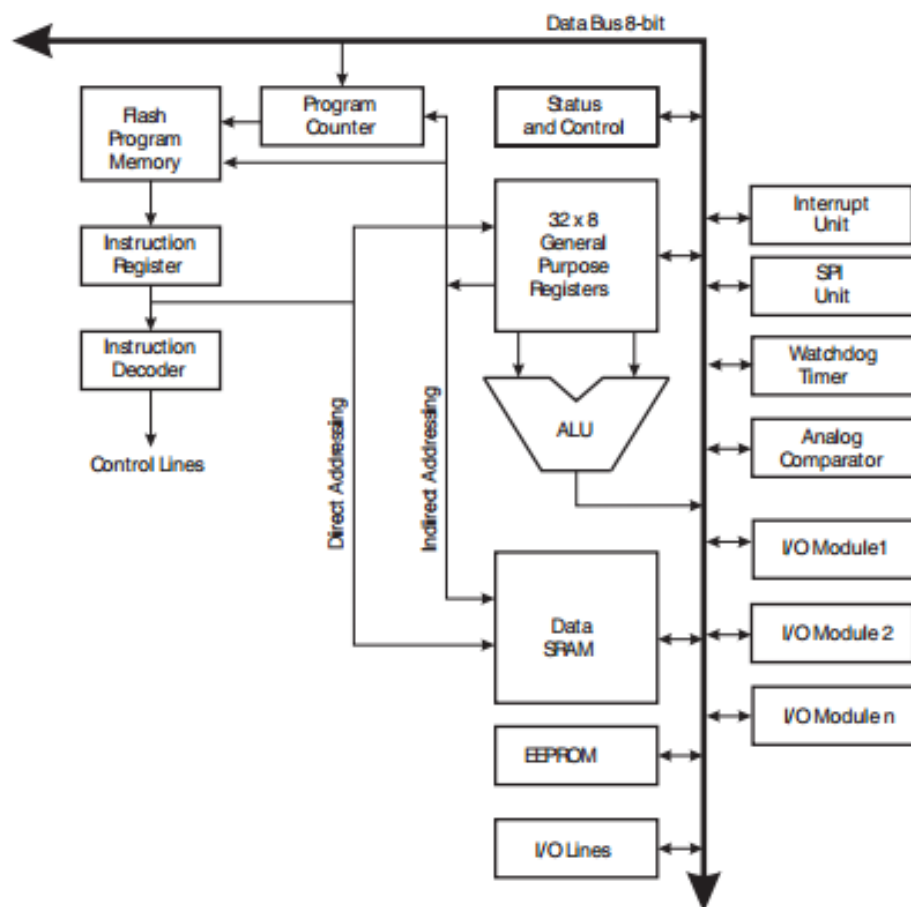


Los dispositivos de la familia ATmega difieren en tamaño de memoria y numero de pines, en la siguiente tabla se muestra las características de cada microcontrolador.

Device	Flash	EEPROM	RAM	General Purpose I/O pins	16 bits resolution PWM channels	Serial USARTs	ADC Channels
ATmega640	64KB	4KB	8KB	86	12	4	16
ATmega1280	128KB	4KB	8KB	86	12	4	16
ATmega1281	128KB	4KB	8KB	54	6	2	8
ATmega2560	256KB	4KB	8KB	86	12	4	16
ATmega2561	256KB	4KB	8KB	54	6	2	8

Arquitectura principal de AVR.

La función principal del núcleo del CPU es asegurar la ejecución del programa. El CPU debe poder acceder a memorias, realizar cálculos, controlar periféricos, y manejar interrupciones.



Como se puede observar en el diagrama a bloques internamente el microcontrolador tiene un procesador el cual lo conforma una unidad aritmética y

lógica ALU, registros, contador de programa, registro de instrucción, decodificador de instrucción, buses etc. A su vez contiene dentro del mismo encapsulado puertos de entrada y salida y sección de memoria, una tipo RAM, tipo ROM con la EEPROM y memoria flash.

Conjunto de instrucciones

El set de instrucciones del ATmega se puede dividir según sus especialidades:

- Instrucciones Aritméticas y lógicas.
- Instrucciones de salto.
- Instrucciones de Transferencia de datos.
- Instrucciones orientadas a Bits.
- Instrucciones de control del MCU.

Instrucciones aritméticas y lógicas

Mnemónicos	Operandos	Descripción	Operación	Banderas afectadas	Ciclos de Reloj
ADD	Rd, Rr	Suma dos Registros	$Rd \leftarrow Rd + Rr$	Z,C,N,V,H	1
ADC	Rd, Rr	Suma dos Registros mas el acarreo	$Rd \leftarrow Rd + Rr + C$	Z,C,N,V,H	1
ADIW	Rdl,K	Suma un dato Inmediato a una Palabra	$Rdh:Rdl \leftarrow Rdh:Rdl + K$	Z,C,N,V,S	2
SUB	Rd, Rr	Resta dos Registros	$Rd \leftarrow Rd - Rr$	Z,C,N,V,H	1
SUBI	Rd, K	Resta una Constante de un Registro	$Rd \leftarrow Rd - K$	Z,C,N,V,H	1
SBC	Rd, Rr	Resta dos Registros y el acarreo	$Rd \leftarrow Rd - Rr - C$	Z,C,N,V,H	1
SBCI	Rd, K	Resta el Acarreo y una Constante de un Registro	$Rd \leftarrow Rd - K - C$	Z,C,N,V,H	1
SBIW	Rdl,K	Resta un dato Inmediato de una Palabra	$Rdh:Rdl \leftarrow Rdh:Rdl - K$	Z,C,N,V,S	2
AND	Rd, Rr	Oper. Log. AND entre dos Registros	$Rd \leftarrow Rd \cdot Rr$	Z,N,V	1
ANDI	Rd, K	Oper. Log. AND entre Registro y una Constante	$Rd \leftarrow Rd \cdot K$	Z,N,V	1
OR	Rd, Rr	Oper. Log. OR entre dos Registros	$Rd \leftarrow Rd \vee Rr$	Z,N,V	1
ORI	Rd, K	Oper. Log. OR entre Registro y una Constante	$Rd \leftarrow Rd \vee K$	Z,N,V	1
EOR	Rd, Rr	Oper. Log. X-OR entre dos Registros	$Rd \leftarrow Rd \oplus Rr$	Z,N,V	1
COM	Rd	Complemento a 1 de un Registro	$Rd \leftarrow 0xFF - Rd$	Z,C,N,V	1
NEG	Rd	Complemento a 2 de un Registro	$Rd \leftarrow 0x00 - Rd$	Z,C,N,V,H	1
SBR	Rd,K	Poner a 1 los bits del Registro	$Rd \leftarrow Rd \vee K$	Z,N,V	1
CBR	Rd,K	Limpiar con 0 los bits del Registro	$Rd \leftarrow Rd \cdot (0xFF - K)$	Z,N,V	1
INC	Rd	Incrementar Registro	$Rd \leftarrow Rd + 1$	Z,N,V	1
DEC	Rd	Decrementar Registro	$Rd \leftarrow Rd - 1$	Z,N,V	1
TST	Rd	Comprobar que un Registro sea Cero o Negativo	$Rd \leftarrow Rd \cdot Rd$	Z,N,V	1
CLR	Rd	Limpiar un Registro	$Rd \leftarrow Rd \oplus Rd$	Z,N,V	1
SER	Rd	Colocar en 1's el Registro	$Rd \leftarrow 0xFF$	Ninguna	1
MUL	Rd, Rr	Multipliación sin signo	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULS	Rd, Rr	Multipliación con signo	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
MULSU	Rd, Rr	Multiplicar un Reg. con signo con uno sin signo	$R1:R0 \leftarrow Rd \times Rr$	Z,C	2
FMUL	Rd, Rr	Multipliación sin signo Fraccionaria	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULS	Rd, Rr	Multipliación con signo Fraccionaria	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2
FMULSU	Rd, Rr	Mult. Fraccionaria de un Reg. con signo con uno sin signo	$R1:R0 \leftarrow (Rd \times Rr) \ll 1$	Z,C	2

Instrucciones de salto

Mnemónicos	Operandos	Descripción	Operación	Banderas afectadas	Ciclos de Reloj
RJMP	k	Salto relativo	$PC \leftarrow PC + k + 1$	Ninguna	2
JMP		Salto Indirecto a (Z)	$PC \leftarrow Z$	Ninguna	2
RCALL	k	Llamado a Subrutina relativa	$PC \leftarrow PC + k + 1$	Ninguna	3
ICALL		Llamada Indirecta a (Z)	$PC \leftarrow Z$	Ninguna	3
RET		Retorno de una Subrutina	$PC \leftarrow STACK$	Ninguna	4
RETI		Retorno de una Interrupción	$PC \leftarrow STACK$	I	4
CPSE	Rd,Rr	Compara y Salta si es igual	if (Rd = Rr) $PC \leftarrow PC + 2$ or 3	Ninguna	1 / 2 / 3
CP	Rd,Rr	Compara	$Rd - Rr$	Z, N,V,C,H	1
CPC	Rd,Rr	Compara con el Acarreo	$Rd - Rr - C$	Z, N,V,C,H	1
CPI	Rd,K	Compara Registro con un dato Inmediato	$Rd - K$	Z, N,V,C,H	1
SBRC	Rr, b	Salta si el Bit en el Reg. es Cero	if (Rr(b)=0) $PC \leftarrow PC + 2$ or 3	Ninguna	1 / 2 / 3
SBRCS	Rr, b	Salta si el Bit en el Reg. es Uno	if (Rr(b)=1) $PC \leftarrow PC + 2$ or 3	Ninguna	1 / 2 / 3
SBIC	P, b	Salta si el Bit en el Reg. de I/O es Cero	if (P(b)=0) $PC \leftarrow PC + 2$ or 3	Ninguna	1 / 2 / 3
SBIS	P, b	Salta si el Bit en el Reg. de I/O es Uno	if (P(b)=1) $PC \leftarrow PC + 2$ or 3	Ninguna	1 / 2 / 3
BRBS	s, k	Salta si la bandera de estado es Uno	if (SREG(s) = 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRBC	s, k	Salta si la bandera de estado es Cero	if (SREG(s) = 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BREQ	k	Salta si es Igual	if (Z = 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRNE	k	Salta si no es Igual	if (Z = 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRCS	k	Salta si el Acarreo esta en Uno	if (C = 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRCC	k	Salta si el Acarreo esta en Cero	if (C = 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRSH	k	Salta si es Igual o Mayor	if (C = 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRLO	k	Salta si es Menor	if (C = 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRMI	k	Salta si es Negativo	if (N = 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRPL	k	Salta si es Positivo	if (N = 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRGE	k	Salta si es Mayor o Igual, con signo	if (N \oplus V= 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRLT	k	Salta si es Menor que Cero, con signo	if (N \oplus V= 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRHS	k	Salta si el Acarreo Intermedio es Uno	if (H = 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRHC	k	Salta si el Acarreo Intermedio es Cero	if (H = 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRTS	k	Salta si la bandera T esta en Uno	if (T = 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRTC	k	Salta si la bandera T esta en Cero	if (T = 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRVS	k	Salta si la bandera de Sobreflujo esta en Uno	if (V = 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRVC	k	Salta si la bandera de Sobreflujo esta en Cero	if (V = 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRIE	k	Salta si la Interrupción está habilitada	if (I = 1) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2
BRID	k	Salta si la Interrupción No está habilitada	if (I = 0) then $PC \leftarrow PC + k + 1$	Ninguna	1 / 2

Instrucciones de transferencia de datos

Mnemónicos	Operandos	Descripción	Operación	Banderas afectadas	Ciclos de Reloj
MOV	Rd, Rr	Movimiento entre Registros	$Rd \leftarrow Rr$	Ninguna	1
MOVW	Rd, Rr	Copia Registro de una Palabra	$Rd+1:Rd \leftarrow Rr+1:Rr$	Ninguna	1
LDI	Rd, K	Carga dato Inmediato	$Rd \leftarrow K$	Ninguna	2
LD	Rd, X	Carga dato Indirecto	$Rd \leftarrow (X)$	Ninguna	2
LD	Rd, X+	Carga dato Indirecto y se Incrementa	$Rd \leftarrow (X), X \leftarrow X + 1$	Ninguna	2
LD	Rd, -X	Se Decrementa y Carga dato Indirecto	$X \leftarrow X - 1, Rd \leftarrow (X)$	Ninguna	2
LD	Rd, Y	Carga dato Indirecto	$Rd \leftarrow (Y)$	Ninguna	2
LD	Rd, Y+	Carga dato Indirecto y se Incrementa	$Rd \leftarrow (Y), Y \leftarrow Y + 1$	Ninguna	2
LD	Rd, -Y	Se Decrementa y Carga dato Indirecto	$Y \leftarrow Y - 1, Rd \leftarrow (Y)$	Ninguna	2
LDD	Rd, Y+q	Carga dato Indirecto desplazado	$Rd \leftarrow (Y + q)$	Ninguna	2
LD	Rd, Z	Carga dato Indirecto	$Rd \leftarrow (Z)$	Ninguna	2
LD	Rd, Z+	Carga dato Indirecto y se Incrementa	$Rd \leftarrow (Z), Z \leftarrow Z+1$	Ninguna	2
LD	Rd, -Z	Se Decrementa y Carga dato Indirecto	$Z \leftarrow Z - 1, Rd \leftarrow (Z)$	Ninguna	2
LDD	Rd, Z+q	Carga dato Indirecto desplazado	$Rd \leftarrow (Z + q)$	Ninguna	2
LDS	Rd, k	Carga dato Directo de la SRAM	$Rd \leftarrow (k)$	Ninguna	2
ST	X, Rr	Almacenamiento Indirecto	$(X) \leftarrow Rr$	Ninguna	2
ST	X+, Rr	Almacena un dato Indirecto y se Incrementa	$(X) \leftarrow Rr, X \leftarrow X + 1$	Ninguna	2
ST	-X, Rr	Se Decrementa y almacena un dato Indirecto	$X \leftarrow X - 1, (X) \leftarrow Rr$	Ninguna	2
ST	Y, Rr	Almacenamiento Indirecto	$(Y) \leftarrow Rr$	Ninguna	2
ST	Y+, Rr	Almacena un dato Indirecto y se Incrementa	$(Y) \leftarrow Rr, Y \leftarrow Y + 1$	Ninguna	2
ST	-Y, Rr	Se Decrementa y almacena un dato Indirecto	$Y \leftarrow Y - 1, (Y) \leftarrow Rr$	Ninguna	2
STD	Y+q, Rr	Se almacena dato Indir. con desplazamiento	$(Y + q) \leftarrow Rr$	Ninguna	2
ST	Z, Rr	Almacenamiento Indirecto	$(Z) \leftarrow Rr$	Ninguna	2
ST	Z+, Rr	Almacena un dato Indirecto y se Incrementa	$(Z) \leftarrow Rr, Z \leftarrow Z + 1$	Ninguna	2
ST	-Z, Rr	Se Decrementa y almacena un dato Indirecto	$Z \leftarrow Z - 1, (Z) \leftarrow Rr$	Ninguna	2
STD	Z+q, Rr	Se almacena dato Indir. con desplazamiento	$(Z + q) \leftarrow Rr$	Ninguna	2
STS	k, Rr	Almacenamiento Directo a la SRAM	$(k) \leftarrow Rr$	Ninguna	2
LPM		Carga Memoria de Programa	$R0 \leftarrow (Z)$	Ninguna	3
LPM	Rd, Z	Carga Memoria de Programa	$Rd \leftarrow (Z)$	Ninguna	3
LPM	Rd, Z+	Carga Memoria de Programa y se Incrementa	$Rd \leftarrow (Z), Z \leftarrow Z+1$	Ninguna	3
SPM		Se almacena Memoria de Programa	$(Z) \leftarrow R1:R0$	Ninguna	-
IN	Rd, P	Lee del Puerto un dato	$Rd \leftarrow P$	Ninguna	1
OUT	P, Rr	Saca un dato por el Puerto	$P \leftarrow Rr$	Ninguna	1
PUSH	Rr	Empuja un registro al Stack	$STACK \leftarrow Rr$	Ninguna	2
POP	Rd	Regresa un registro del Stack	$Rd \leftarrow STACK$	Ninguna	2

Instrucciones orientadas a manipulación de bits

Mnemónicos	Operandos	Descripción	Operación	Banderas afectadas	Ciclos de Reloj
SBI	P, b	Poner a 1 el Bit de Registro de I/O	$I/O(P, b) \leftarrow 1$	Ninguna	2
CBI	P, b	Poner a 0 el Bit de Registro de I/O	$I/O(P, b) \leftarrow 0$	Ninguna	2
LSL	Rd	Corrimiento Logico a la Izquierda	$Rd(n+1) \leftarrow Rd(n), Rd(0) \leftarrow 0$	Z, C, N, V	1
LSR	Rd	Corrimiento Logico a la Derecha	$Rd(n) \leftarrow Rd(n+1), Rd(7) \leftarrow 0$	Z, C, N, V	1
ROL	Rd	Rotar a la Izquierda a través del Acarreo	$Rd(0) \leftarrow C, Rd(n+1) \leftarrow Rd(n), C \leftarrow Rd(7)$	Z, C, N, V	1
ROR	Rd	Rotar a la Derecha a través del Acarreo	$Rd(7) \leftarrow C, Rd(n) \leftarrow Rd(n+1), C \leftarrow Rd(0)$	Z, C, N, V	1
ASR	Rd	Corrimiento a la Derecha Aritmético	$Rd(n) \leftarrow Rd(n+1), n=0..6$	Z, C, N, V	1
SWAP	Rd	Intercambiar Nibbles	$Rd(3..0) \leftarrow Rd(7..4), Rd(7..4) \leftarrow Rd(3..0)$	Ninguna	1
BSET	s	Poner a 1 las banderas	$SREG(s) \leftarrow 1$	SREG(s)	1
BCLR	s	Poner a 0 las banderas	$SREG(s) \leftarrow 0$	SREG(s)	1
BST	Rr, b	Almacenar Bit de un Register a T	$T \leftarrow Rr(b) \ T \ 1$	T	1
BLD	Rd, b	Cargar Bit de T al Registro	$Rd(b) \leftarrow T$	T	1
SEC		Poner a 1 el Acarreo	$C \leftarrow 1$	C	1
CLC		Poner a 0 el Acarreo	$C \leftarrow 0$	C	1
SEN		Poner a 1 la bandera de los Negativos	$N \leftarrow 1$	N	1
CLN		Poner a 0 la bandera de los Negativos	$N \leftarrow 0$	N	1
SEZ		Poner a 1 la bandera de Ceros	$Z \leftarrow 1$	Z	1
CLZ		Poner a 0 la bandera de Ceros	$Z \leftarrow 0$	Z	1
SEI		Habilitar la Interrupción Global	$I \leftarrow 1$	I	1
CLI		No Habiliar la Interrupción Global	$I \leftarrow 0$	I	1
SES		Poner 1 la band. Comprobación de Signo	$S \leftarrow 1$	S	1
CLS		Poner 0 la band. Comprobación de Signo	$S \leftarrow 0$	S	1
SEV		Poner 1 al Sobreflujo del Complemento a 2	$V \leftarrow 1$	V	1
CLV		Poner 0 al Sobreflujo del Complemento a 2	$V \leftarrow 0$	V	1
SET		Poner a 1 la bandera T en SREG	$T \leftarrow 1$	T	1
CLT		Poner a 0 la bandera T en SREG	$T \leftarrow 0$	T	1
SEH		Poner a 1 el Acarreo Auxiliar en SREG	$H \leftarrow 1$	H	1
CLH		Poner a 0 el Acarreo Auxiliar en SREG	$H \leftarrow 0$	H	1

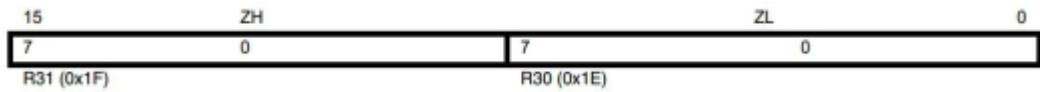
Instrucciones de control de MCU

Mnemónicos	Operandos	Descripción	Operación	Banderas afectadas	Ciclos de Reloj
NOP		No Operacion		Ninguna	1
SLEEP		Sleep		Ninguna	1
WDR		Watchdog Reset		Ninguna	1
BREAK		Break For On-chip Debug Only		Ninguna	N/A

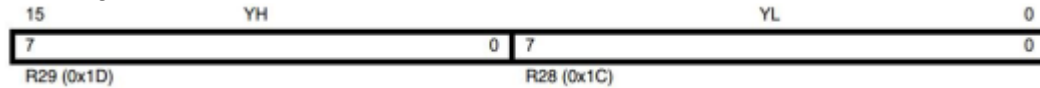
Desarrollo

Escriba un programa que invierte la posición de bits, intercambiando el orden de bits del más significativo al menos significativo, del valor dado en:

I. El registro R31-R30 si R24 es 0,



II. El registro R29-R28 si R24 es 1,



III. El programa realizará ninguna acción para cualquier otro valor en R24.

Nota: El programa debe realizar estas acciones indefinidamente.

Conclusiones y comentarios:

La arquitectura del microcontrolador ATmega en sus diferentes versiones contienen lo básico de un sistema computacional para trabajar como son unidad de procesamiento, sección de memoria y sección de entrada y salida de manera muy limitada, esto porque un microcontrolador funciona para una tarea en específico o están orientados a realizar una tarea o pocas, en contraste que el microprocesador es de uso general.

El lenguaje ensamblador de AVR es muy parecido a lo que ya conocía (x86) y no me tomo mucho trabajo o no se me presento mucha dificultad al realizar la practica por el gran parecido a lo que ya conozco de programación a nivel ensamblador. Sera agradable aprender y trabajar con estos dispositivos.

Bibliografía:

[1] Modos de Direcccionamiento e Instrucciones del AVR ATmega32. 30-03-2017, de ITPS Sitio web:

https://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwjrsObmP_SAhVml1QKHb7xAAUQFggcMAE&url=https%3A%2F%2Fvirtual.unet.edu.ve%2Fpluginfile.php%2F199032%2Fmod_folder%2Fcontent%2F0%2FTutorial%2520ATMEGA%2F7842870-Capitulo3-Modos-de-Direcccionamiento-e-Instrucciones-del-ATmega32-espanol.pdf%3Fforcedownload%3D1&usg=AFQjCNE8Y2Y8nnAxwPVNRMwKdbRw2Mf4BA&sig2=K4YtBYHdc3GtKr0yWwcRxA&bvm=bv.151325232,d.cGw