



# Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería

## Microprocesadores y Microcontroladores

Práctica 6:  
"Manejo de la sección E/S del microcontrolador  
ATmega1280/2560"

Chávez Padilla Ignacio  
1246720

Grupo : 562

Jesús García

Viernes, 12 de Abril de 2019

## Objetivo

Mediante esta práctica el alumno analizará la implementación de retardos por software, así como también se familiarizará con la configuración y uso de puertos.

## Equipo

- Computadora Personal

•

## Teoría

### Ensamblador en linea gcc

El compilador de C de GNU permite mezclar el lenguaje ensamblador dentro de programas en C. Esta opción se puede utilizar para manualmente optimizar partes críticas del software o para utilizar una instrucción muy específica del procesador, que puede no estar disponible en el lenguaje C. Una instrucción simple puede ser la siguiente:

```
asm("in %0, %1" : "=r" (value) : "I" (_SFR_IO_ADDR(PORTD)))
```

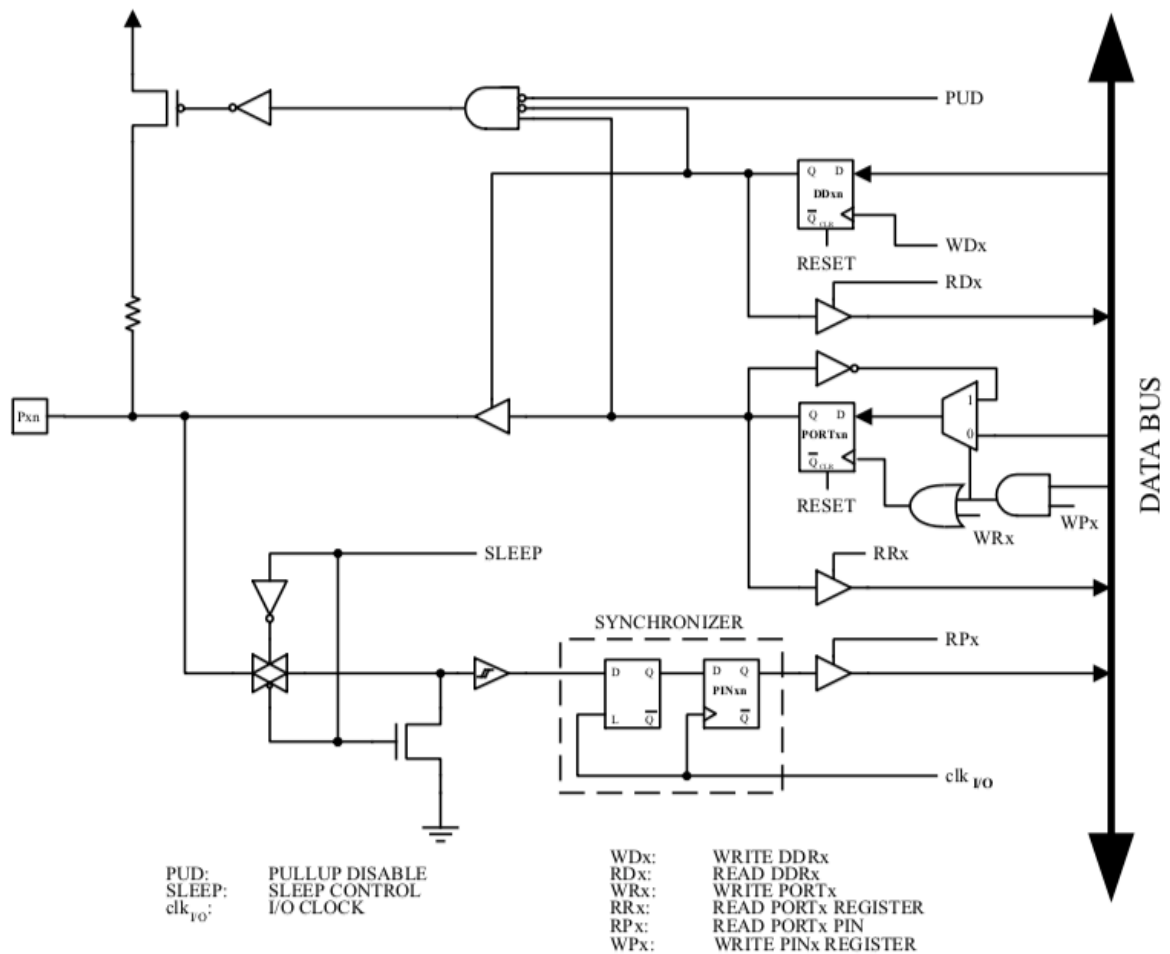
Se divide en cuatro partes:

1. Las instrucciones de ensamblador, definidas como una cadena: "in %0, %1"
2. Las salidas: "=r"
3. Una lista de operandos de entrada separados por comas: "I" (\_SFR\_IO\_ADDR(PORTD))
4. Registros afectados (vacíos en este caso)

### Puertos de E/S en ATmega 1280/2560

Los puertos de E/S del microcontrolador tienen verdadera funcionalidad de Escribir-Modificar-Leer cuando se utilizan como puertos digitales generales. Lo que significa que la dirección de un solo bit de un puerto puede ser cambiado sin intencionalmente intercambiar la dirección de cualquier otro pin con las instrucciones de SBI y CBI. Lo mismo aplica al modificar el valor de la dirección o la habilitación de resistores pull up internos en caso de que estos sean configurados como entrada.

Figure 15-2. General Digital I/O<sup>(1)</sup>



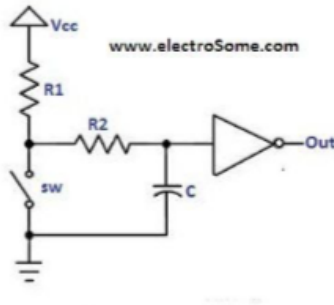
Cada pin del puerto consiste de tres bits de registro.

- **$DD_{xn}$**  afecta la dirección del pin, si es uno,  $P_{xn}$  es configurado como salida. Sí  $DD_{xn}$  es cero,  $P_{xn}$  es configurado como entrada.
- **$P_{xn}/PORT_{xn}$**  es escrito un uno al ser de entrada, el resistor pull-up se activa. Para desactivarlo, se tiene que escribir un cero.. Los puertos se convierten en tri-estado(Hi-Z) cuando se activa un reset. Si es escrito un uno cuando es configurado como pin de salida, la salida es de uno, de ser cero la salida es cero.
- **$PIN_{xn}$**  regresa los valores de entrada en las entradas de  $PORT_{xn}$ . Esto permite que se puedan utilizar configuraciones diferentes de los pines en un solo puerto, dando flexibilidad.

## Técnicas anti-rebote(debounce)

- Mediante hardware Resistor Capacitor

Este circuito es utilizado para eliminar el rebote con un capacitor. Cuando el switch está abierto, el voltaje a través del capacitor que es inicialmente cero ahora se dirige al resistor R1 y R2.



El voltaje en Vin es alto por lo que el disparador Schmitt es bajo. Si el switch está cerrado el capacitor se descarga y el voltaje en Vin es 0, por lo que el disparador es 1.

- Mediante software

Esta solución involucra el activar un procedimiento al momento de presionar el botón por software, justo después de entrar se tiene que realizar un delay, por lo que se ignoran las señales del rebote durante este periodo de tiempo (en milisegundos), al momento de que se acabe este delay. Se puede revisar de nuevo el estado del botón en caso de ser necesario.

## Conclusiones y Comentarios

Se aprendió como configurar y usar los puertos del Atmega2560, así mismo se reforzó el conocimiento adquirido en clase anteriores sobre técnicas anti rebote.

Es importante recordar que las técnicas anti rebote pueden llevarse a cabo por software o hardware, en esta practica se realizo mediante software.

## Bibliografía

[https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561\\_datasheet.pdf](https://ww1.microchip.com/downloads/en/devicedoc/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf)

Atmel ATmega640/V-1280/V-1281/V-2560/V-2561/V