



Departamento de Electrónica
Laboratorio de Microprocesadores - 86.07

Trabajo Práctico Obligatorio N°2: Manejo de Puertos

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1°/2020									
Turno de las clases prácticas			Miércoles									
Jefe de trabajos prácticos:			Ing. Pedro Ignacio Martos									
Docente guía:			Ing. Fabricio Baglivo, Ing. Fernando Pucci									
Autor			Seguimiento del proyecto									
Maximiliano	Porta	98800										

Observaciones:

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

19 de junio de 2020

Índice

1. Objetivo	2
2. Descripción	2
2.1. Introducción	2
2.2. Desarrollo	3
2.3. Diagrama de Conexión del Circuito Físico	4
2.4. Esquema en Bloques	5
2.5. Materiales Utilizados	6
2.6. Diagrama de Flujo	6
2.7. Lógica del Programa	6
2.8. Links de funcionamiento del circuito:	7
3. Código	7
3.1. Código Sin Resistencia de Pull-Up	7
3.2. Código con Resistencia de Pull-Up	7
4. Resultados y Conclusiones	8
4.1. Resultados	8
4.2. Conclusiones	8

1. Objetivo

Consiste en aprender mas código del lenguaje ensamblador, aprender a configurar las entradas y salidas del microcontrolador, ver el cambio entre usar solo el pulsador y agregar una resistencia de pull-up, tener en cuenta la corriente máxima que puede manejar por pin y máxima corriente de salida total del dispositivo.

2. Descripción

2.1. Introducción

Para el desarrollo de este trabajo de laboratorio se utilizo el diagrama de la Figura 1 y el diagrama de la figura 2. Se observo en la hoja de datos del Arduino la corriente máxima de entrada/salida ($I = 20mA$), usando las leyes de ohm se obtiene la ecuación 1, con una Resistencia de 220Ω y la tensión para un diodo led verde, se obtuvo que la corriente de salida es $I_{Out} = 10,9mA$, por lo cual es menor a la máxima permitida por pin del arduino. Mediante la programación se busca poder controlar mediante una señal externa en este caso un pulsador, la activación de un puerto de salida en este caso encender un diodo led.

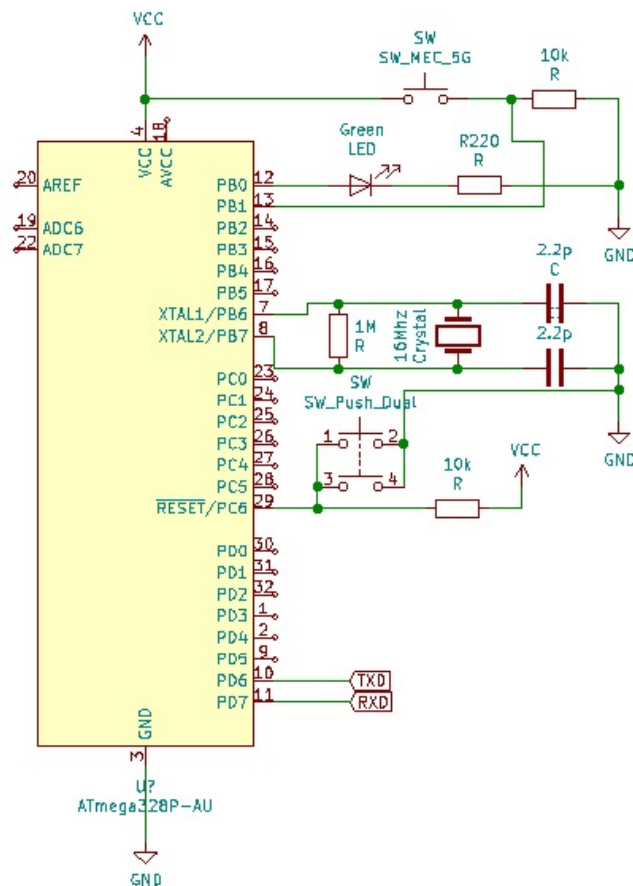


Figura 1: Diagrama de conexión sin resistencia de Pull-Up

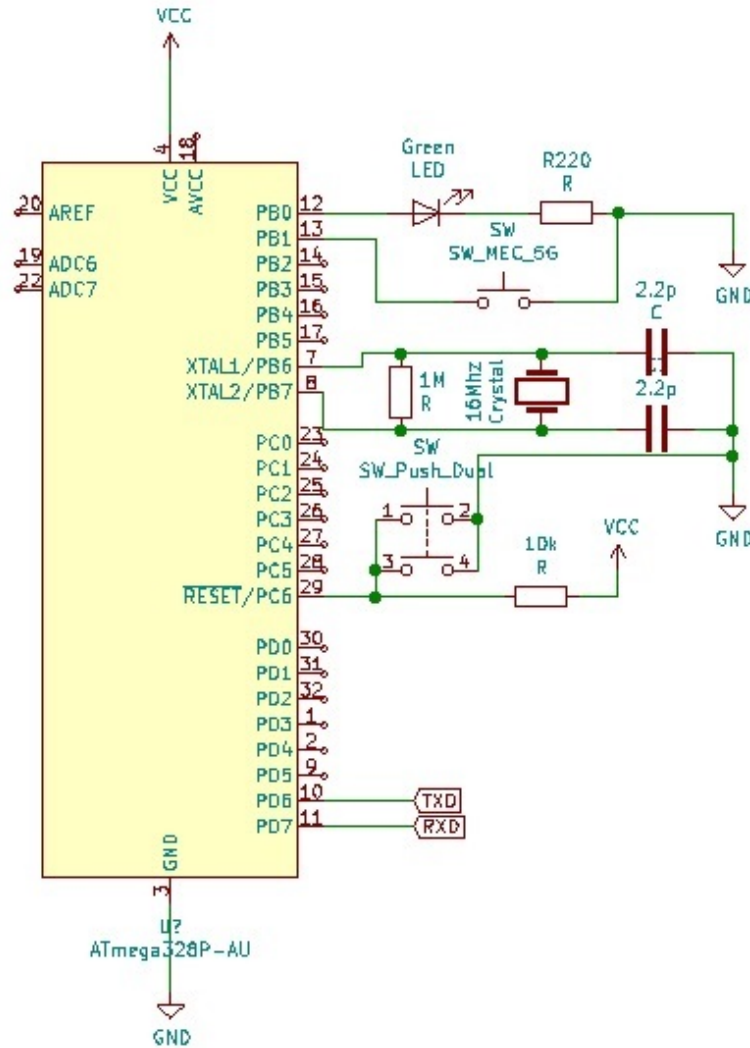


Figura 2: Diagrama de conexión con resistencia de Pull-Up

$$I_{Led} = \frac{V_{CC} - V_{Led}}{R_{Led}} \quad (1)$$

2.2. Desarrollo

Como se observa en la figura 1, en este caso no se usa la resistencia interna de pull-up, por lo tanto se conecta el pulsador a Vcc y se tiene un resistor de $10K\Omega$ para no exceder la corriente máxima de entrada del dispositivo, en este caso entrara un uno lógico. Para la configuración con la resistencia interna de pull-up como se ve en la figura 2 y como se observa en la figura 3, acá la lógica cambia un poco por que la resistencia de pull-up ya esta alimentada con una tensión positiva por por eso para cerrar el circuito tengo que conectar el pulsador a Gnd, su habilitación depende de valor del flip-flop PORTxn, del flip-flop DDRxn y del bit PUD (PUD: Pull-Up Disable). El bit PUD se encuentra en el Registro I/O denominado SFIOR (SFIOR: Special Function I/O Register, Registro I/O de función especial). Con estos 3 bits configurado DDRxn en 0, PORTxn en 1 y PUD en 0. Esto garantiza un 1 lógico cuando el dispositivo está abierto. El otro extremo del botón o interruptor se conecta a tierra, de manera que cuando se cierra el circuito introduce un 0 lógico, el resistor de Pull-Up evita un corto entre el voltaje de alimentación y tierra.

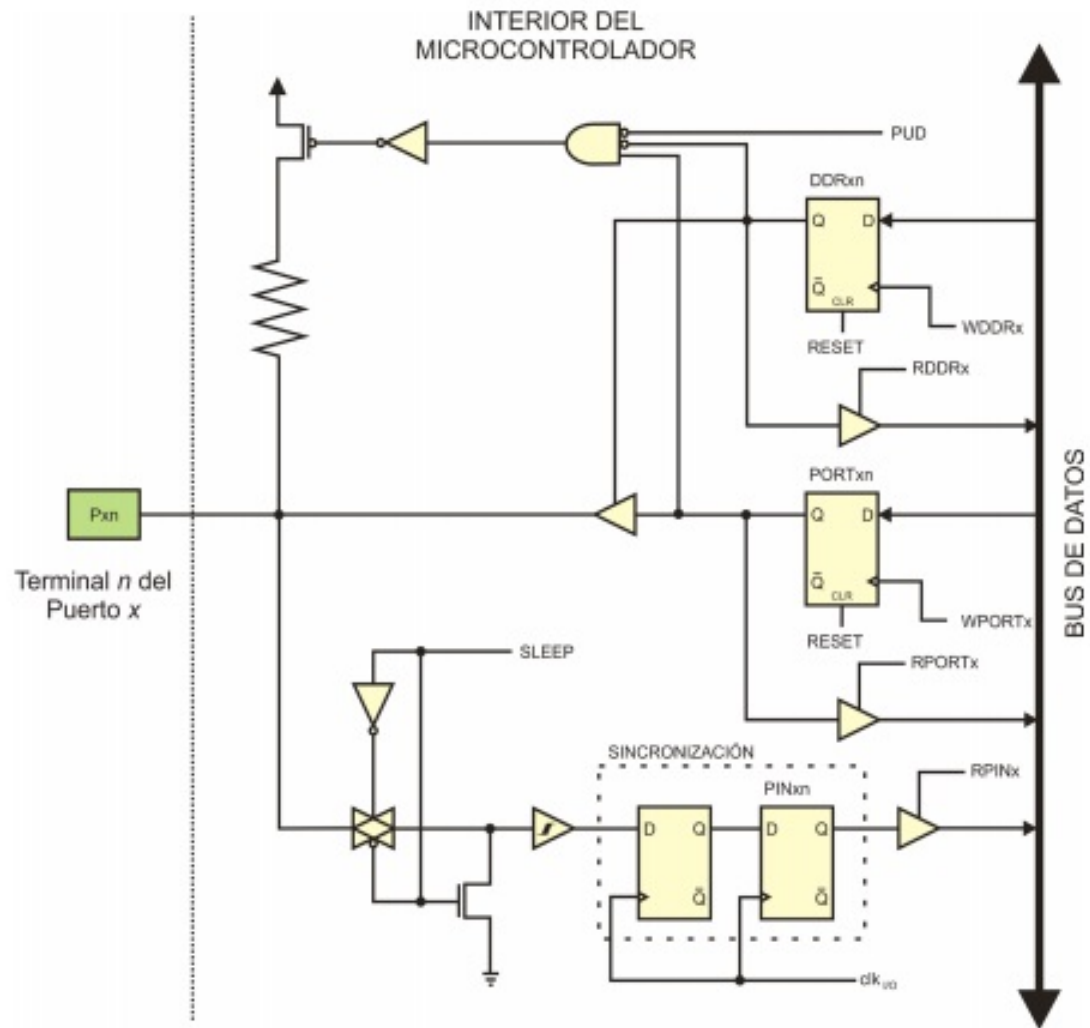


Figura 3: Diagrama de interior del microcontrolador

2.3. Diagrama de Conexión del Circuito Físico

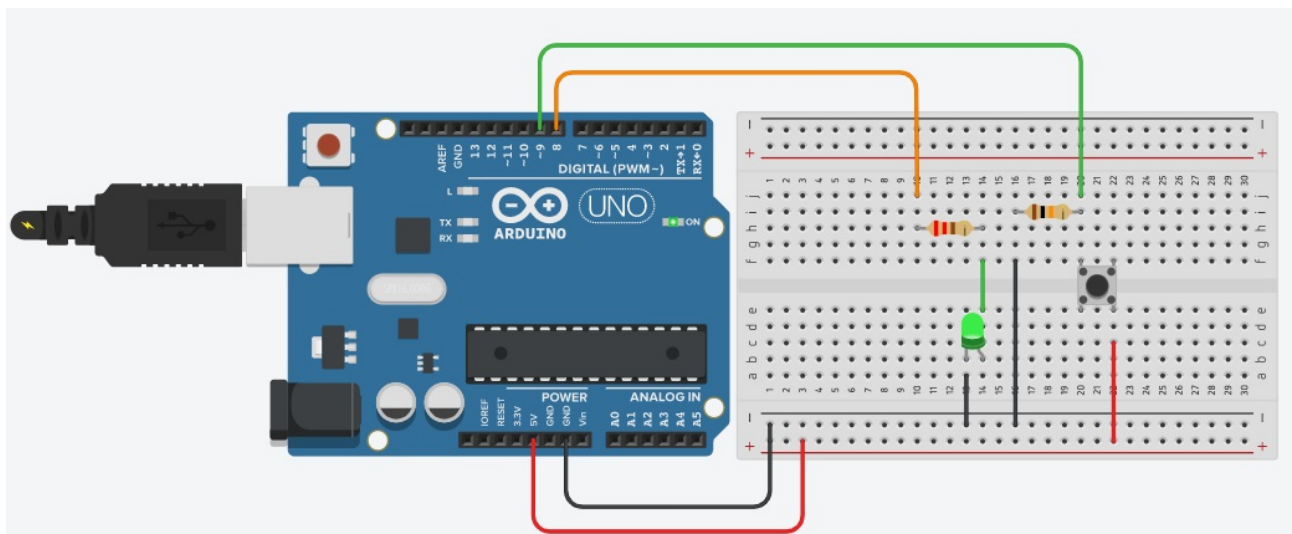


Figura 4: Diagrama del Circuito Sin Resistencia de Pull-Up

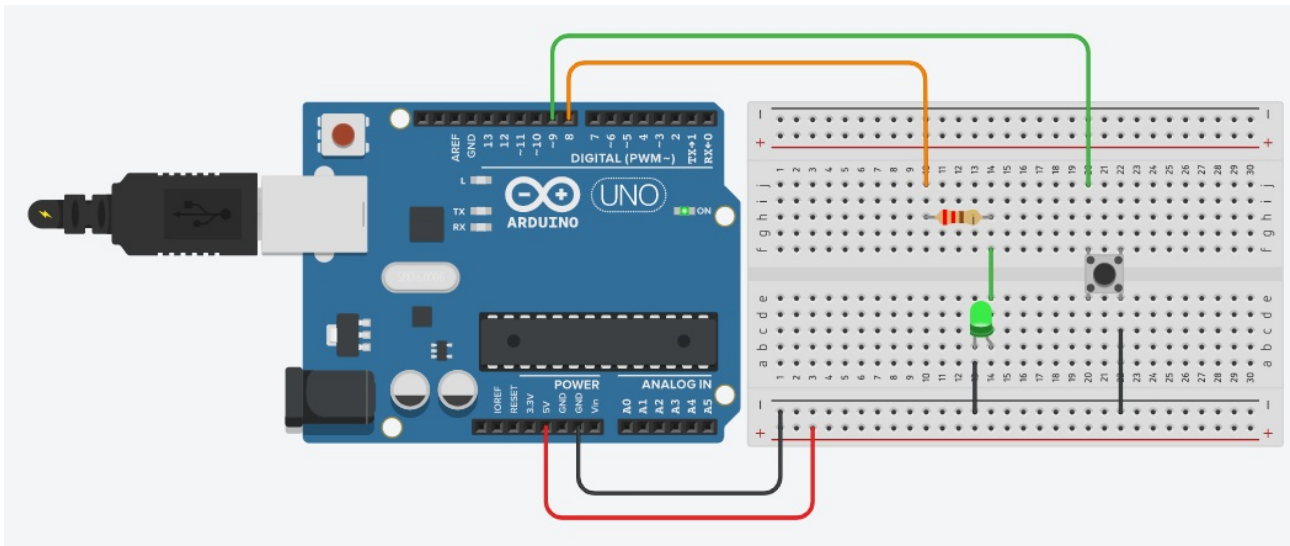


Figura 5: Diagrama del Circuito Con Resistencia de Pull-Up

2.4. Esquema en Bloques

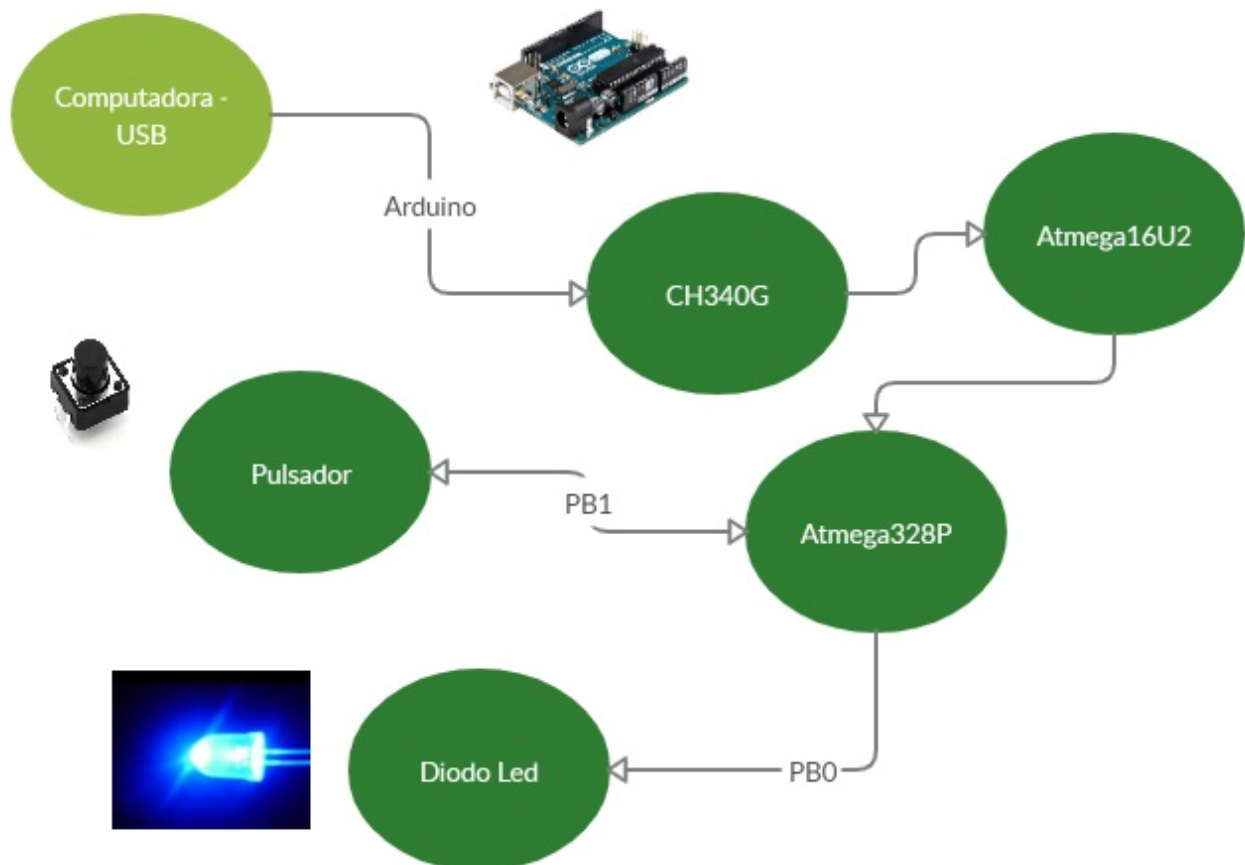


Figura 6: Diagrama en bloques

2.5. Materiales Utilizados

- x1(2.5 Ars) Pulsador de 2 o 4 pines
- x1(6.5 Ars) Diodo Led verde de 3mm (Alta Luminosidad).
- x1(1.05 Ars) Resistor de 200Ω de carbón 1/4 Watt.
- x1(1.05 Ars) Resistor de $10K\Omega$ de carbón 1/4 Watt.
- 1x(725 Ars) Placa Arduino Uno con USB.
- 1x(96.5 Ars) Placa Experimental.
- 1x(10 Ars) Cables de conexión.
- Costo del Proyecto : 845 Ars

2.6. Diagrama de Flujo

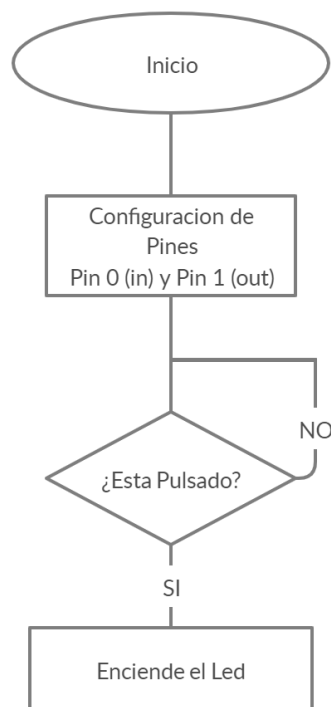


Figura 7: Diagrama de flujo

2.7. Lógica del Programa

Usando la función SBIC le paso el puerto de entrada si esta en 0, no esta pulsado, si esto se cumple saltea la instrucción siguiente, pero si esto es falso quiere decir que se estaría pulsando, entonces ejecuto la instrucción siguiente que es activar la salida del led. Si la función SBIC fue verdadera ahora salto a SBIS, con esta veo si esta pulsado, en caso de ser falso ejecutaría la siguiente instrucción que es apagar la salida, si esta fue verdadera hago un JMP y vuelvo a SBIC y acá es un loop infinito.

2.8. Links de funcionamiento del circuito:

Sin Resistencia de Pull-Up: <https://youtu.be/mI6EXZ8JFFg>.

Con Resistencia de Pull-Up: <https://youtu.be/JEb4aVhNEUo>

3. Código

3.1. Código Sin Resistencia de Pull-Up

```
; Facultad de Ingenieria – UBA
; Laboratorio de Microprocesadores
; Pulsador.asm
; Created: 23/5/2020 14:41:40
; Author : Porta, Maximiliano Adrian
;

; Defino constantes
.EQU ENTRADA = PINB
.EQU SALIDA = PORTB
.EQU PULSADOR_ON = 1
.EQU LED = 0

.CSEG

        JMP MAIN
.ORG INT_VECTORS_SIZE

MAIN:
        CLR R20
        LDI R20, 0b11111101
        OUT DDRB, R20

CICLO_INFINITO:

        SBIC ENTRADA,PULSADOR_ON
        ; Si NO estoy pulsando no se ejecuta lo de abajo
        SBI  SALIDA,LED                      ; Enciende el led
        SBIS ENTRADA,PULSADOR_ON
        ; Si estoy pulsando no ejecuta lo de abajo
        CBI  SALIDA,LED                      ; Apaga el led
        JMP  CICLO_INFINITO
```

3.2. Código con Resistencia de Pull-Up

```
; Facultad de Ingenieria – UBA
; Laboratorio de Microprocesadores
; Pullup.asm
; Created: 23/5/2020 17:06:46
; Author : Porta, Maximiliano Adrian
;
```



```

; Defino constantes
.EQU ENTRADA = PINB
.EQU SALIDA = PORTB
.EQU PULSADOR_ON = 1
.EQU LED = 0

.CSEG

        JMP MAIN
.ORG INT_VECTORS_SIZE

MAIN:
    CLR R20
    LDI R20, 0b11111101
    ; pin 0 como salida y pin 1 como entrada y el resto como salidas
    OUT DDRB, R20
    SBI PORTB, 1      ; Activo la resistencia de Pull_Up

CICLO_INFINITO:

    SBIC ENTRADA, PULSADOR_ON
    ; Si NO estoy pulsando no se ejecuta lo de abajo
    CBI SALIDA, LED      ; Apaga el led
    SBIS ENTRADA, PULSADOR_ON      ; Si estoy pulsando no ejecuta lo de
    SBI SALIDA, LED      ; Enciende el led
    JMP CICLO_INFINITO

```

4. Resultados y Conclusiones

4.1. Resultados

Se logro controlar un pin de salida, mediante un pulsador a la entrada del microcontrolador y se probo como funciona internamente los resistores de pull-up que trae integrado el microcontrolador.

4.2. Conclusiones

Aprendí a utilizar las entradas y salidas del microcontrolador y a controlar las salidas del mismo en función de las entradas en este caso con un pulsador. También aprendí el funcionamiento interno de las resistencias de Pull-Up ya que las mismas permiten controlar al dispositivo con un 0 lógico en vez de un 1 lógico, esto me evita que entre en corto el dispositivo entre Vcc y Gnd, y también a utilizar nuevas funciones en assembler como SBIC y SBIS.