

```

        end if;
    end process conteo;

--Proceso para la salida de datos a los leds
led:process (rst,reg,tmp)
begin
    if rst='1' then leds <= (others=>'0');
--se verifica que ya se hayan mandado los 22 números de trama al soltar la tecla
    elsif tmp = "10110" then
--al soltar la tecla se envía el código de rastreo F0 "break code" y el código de la tecla
    elsif reg(8 downto 1)= X"F0" then leds <= reg(19 downto 12);
    else
        leds <= (others=>'0');
    end if;
end process led;
end Arq_teclado;

```

“Si tienes alguna duda, apóyate con tu profesor”

Desarrollo. **Procedure**

Recuerde **MOSTRAR** sus circuitos funcionando a su profesor para la valoración del trabajo de laboratorio (TL2).

1. Implementar una unidad aritmético-lógica **ALU**, usando diseño de alto nivel, que realice las operaciones aritméticas **A+B**, **A-B**, **AxB**, **A/B** y las operaciones lógicas **not A**, **A and B**, **A or B**, **A xor B**, siendo las entradas A y B de por lo menos 4 bits, con salida aritmética “C” de 4 a 8 bits, empleando un selector “**sel**” de la operación de 3 bits de acuerdo a la tabla 3.1 y la figura 3.1 (repetidas abajo). Por lo menos una de las entradas es un encoder mecánico rotatorio. Un led bicolor o **RGB** en **Rojo** indica una operación aritmética y en **Azul** una operación lógica. El lenguaje en el que se implementa lo define el profesor. Reportar los códigos comentados, simulaciones, fotos del funcionamiento con texto explicativo de lo que sucede.

*Implement an ALU arithmetic-logic unit, using a top level design, that performs the arithmetic operations **A+B**, **A-B**, **AxB**, **A/B** and the logical operations **not A**, **A and B**, **A or B**, **A xor B**, being the inputs A and B of at least 4 bits, with output C of 4 to 8 bits, using a “**sel**” selector of the 3-bit operation according to table 3.1 and figure 3.1 (repeated below). At least one of the inputs is a rotative mechanical encoder. A two-color led or **RGB** in **Red** indicates an arithmetic operation and in **Blue** a logical operation. The language in which it is implemented is defined by the teacher. Report the commented codes, simulations, operation photos with explanatory text of what happens.*

Tabla 3.1. Relación de operaciones de la ALU

Operaciones	selOp	Operación	Observaciones @ A y B de 4 bits
Aritméticas	000	$C=A+B$	C es de 5 bits (máx 30)
Salida a leds y a display.	001	$C=A-B$	C es de 4 bits (de -15 a 15)
	010	$C=A \times B$	C es de 8 bits (máx 225)
Led RGB Rojo.	011	$C=A/B$	C es de 4 bits (solo enteros de 1 a 15)
Lógicas	100	$C=\text{not}(A)$	C es de 4 bits
Salida a leds.	101	$C=A \text{ and } B$	C es de 4 bits
Led RGB Azul.	110	$C=A \text{ or } B$	C es de 4 bits
	111	$C=A \text{ xor } B$	C es de 4 bits

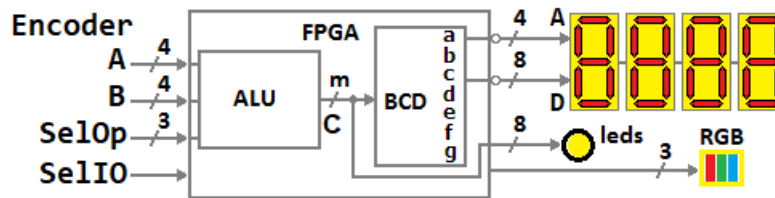


Figura 3.1. Diagrama a bloques de la ALU y su conexión.

Challenge 1 (substitute points 1 and 3).

Implement an ultrasonic distance meter from 0 to 400cm using the HC-05 or SRF-05 or similar, with 7-segment display output. Within the range of 0 to 30cm a proximity alarm is activated, which can be enabled and disabled with a switch. Report the internal parts (dividers, counters, BCD, etc.) with a block diagram, codes (HDL and UCF) and photos.

- Implementar un circuito de control de giro para un **motor a pasos unipolar** con encoder mecánico rotatorio y encoder magnético u óptico, tal que al girar cualquiera de los encoder, la carga mecánica girará en la misma dirección (horario o antihorario) además de encender un led RGB que indique la dirección. Incluir sensores de límite para proteger el giro de la carga mecánica y un indicador audible ("beep") que indique ciertas posiciones. Es posible colocar un selector para direccionar la señal de cada encoder, funcionando de acuerdo a la figura 3.2 (repetida abajo). El motor deberá llevar una **carga** en el eje (banda, llantas, polea, sistema animatrónico, ventana o puerta a escala, etc., no se permiten cosas sencillas como hélices, clips, alambres, ligas, hojas de papel, etc.). El lenguaje en el que se implementa lo define el profesor. Reportar los códigos comentados, fotos del funcionamiento con texto explicativo de lo que sucede.

*Implement a control circuit for a unipolar stepper motor with rotary mechanical encoder and magnetic or optical encoder, such that when turning any of the encoders, the motor will turn in the same direction (clockwise or counterclockwise) as well as turning on an RGB LED that indicate the address. Include limit sensors to protect the mechanical load rotation or an audible indicator ("beep") that indicates certain positions. It is possible to place a selector to address the signal of each encoder, working according to figure 3.2 (repeated below). The motor must have a **load** on the shaft (belt, wheels, pulley, animatronic system, window or door to scale, etc., simple things such as propellers, clips, wires, links, sheets of paper, etc. are not allowed). The language in which it is*

implemented is defined by the teacher. Report the commented codes, simulations, operation photos with explanatory text of what happens

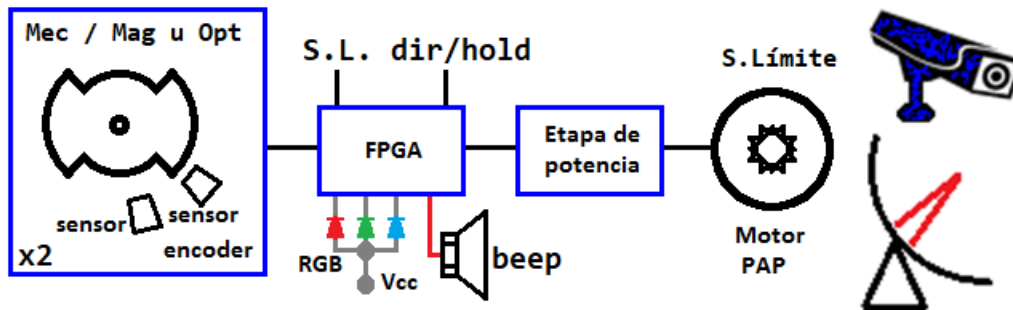


Figura 3.2. Diagrama a bloques del control del motor a pasos con encoder.

- Implementar en TLD un **contador de pulso en alto en ms** con salida a display, el cual al presionar un botón (push) se visualiza el conteo en el display y cuando se suelta el botón se detiene en un valor, pero si se presiona otra vez continua el conteo. Se cuenta con un botón de reset para reiniciar en cero en cualquier momento que se presione. El display puede contar hasta un máximo de 9,999 ms (9.9 s) y cuando se alcance este valor se detendrá y sonará un mensaje en una bocina (“valor máximo”, “ya me llené”, etc.). Por medio de un led RGB se indica en que rango en que se encuentra el conteo ($0 < R \leq 3,333$, $3,333 < G \leq 6,666$, $6,666 < B \leq 9,999$). Cada vez que se cambia de rango sonará un “BEEP” en un buzzer o en la bocina. En la figura 3.3 se muestra el diagrama a bloques, en la 3.18 un ejemplo del funcionamiento y en la 3.19 una forma de hacer el TLD. El lenguaje en el que se implementa lo define el profesor. Reportar los códigos comentados, simulaciones, fotos del funcionamiento con texto explicativo de lo que sucede.

Implement in TLD a pulse counter in high in ms with output to display, which when pressing a button (push) the count is displayed on the display and when the button is released it stops at a value, but if it is pressed again continue the count. It has a reset button to restart at zero any time it is pressed. The display can count to a maximum of 9,999 ms (9.9 s) and when this value is reached it will stop and a message (“maximum value”, “I am already filled”) will sound in a speaker, etc. By means of a RGB led it is indicated in what rank the count is ($0 < R \leq 3,333$, $3,333 < G \leq 6,666$, $6,666 < B \leq 9,999$). Each time the range is changed, a “BEEP” will sound in a buzzer or speaker. The block diagram is shown in figure 3.3, an example of the operation in figure 3.18 and one way to implement the TLD in figure 3.19. The language in which it is implemented is defined by the teacher. Report the commented codes, simulations, operation photos with explanatory text of what happens.

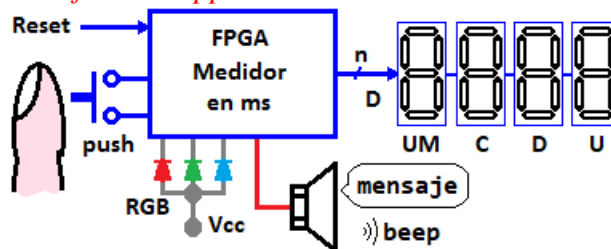


Figura 3.3. Esquema para el medidor de ms.

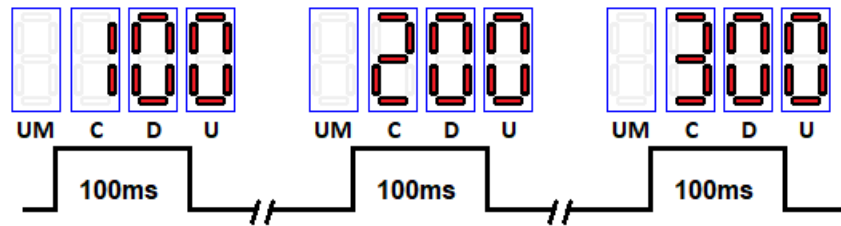


Figura 3.18. Ejemplo de funcionamiento cuando se presiona 3 veces 100ms.

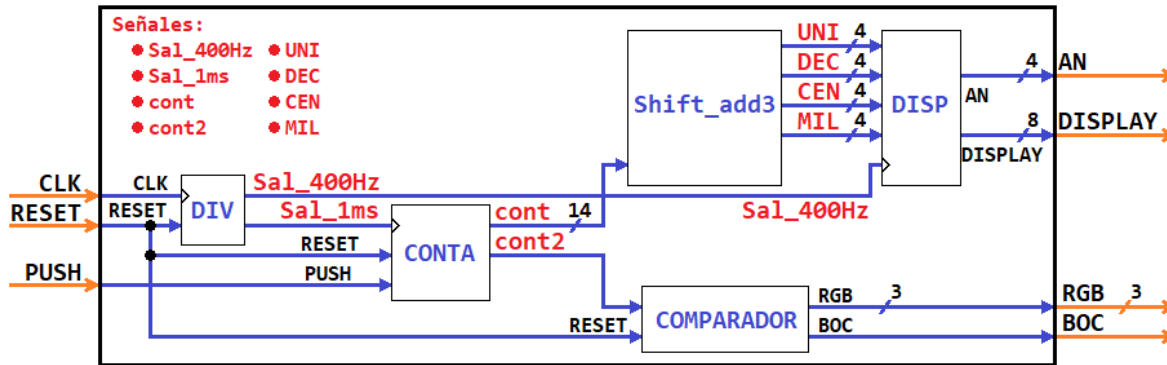


Figura 3.19. Estructura de bloques para el medidor de ms en TLD.

Consultar el boletín UPIITA número 63 (www.boletin.upiita.ipn.mx o <http://www.boletin.upiita.ipn.mx/index.php/component/content/article/11-numeros/1427-numero-63>).

- Implementar un control de velocidad y sentido de giro de un **servomotor (sin tope)** con encoder rotatorio (óptico o magnético, ver figura 3.4). El movimiento del encoder en un sentido permitirá que el motor incremente su velocidad, mientras se le den más vueltas seguirá aumentando, pero si se dan vueltas en sentido contrario la velocidad disminuirá, ya sea hasta detenerse o bien para cambiar el sentido de giro e ir aumentando su velocidad en sentido contrario. En el display (ver figura 3.20) se muestra la velocidad en niveles numéricos (-5 a 5). Se utiliza una señal PWM adecuada al motor adquirido, **con carga** en el eje (banda, llantas, polea, sistema animatrónico, ventana o puerta a escala, etc.), no se permiten cosas sencillas como hélices, clips, alambres, ligas, hojas de papel, etc.). El lenguaje en el que se implementa lo define el profesor. Reportar los códigos comentados, simulaciones, fotos del funcionamiento con texto explicativo de lo que sucede.

*Implement a speed control and direction of rotation of a servomotor (without stop) with rotary encoder (optical or magnetic, see figure 3.4). The movement of the encoder in one direction will allow the engine to increase its speed, as more turns are made in the same direction the encoder will continue to increase, but if they are turned in the opposite direction the speed will decrease, either to stop or to change the direction of rotation and increase its speed in the opposite direction. The display (see figure 3.20) shows the speed in numerical levels (-5 to 5). A suitable PWM signal is used for the motor purchased, **with load** on the shaft (belt, wheels, pulley, animatronic system, window or*

door to scale, etc., simple things such as propellers, clips, wires, rubber bands, paper sheet, etc. are not allowed). The language in which it is implemented is defined by the teacher. Report the commented codes, simulations, operation photos with explanatory text of what happens.

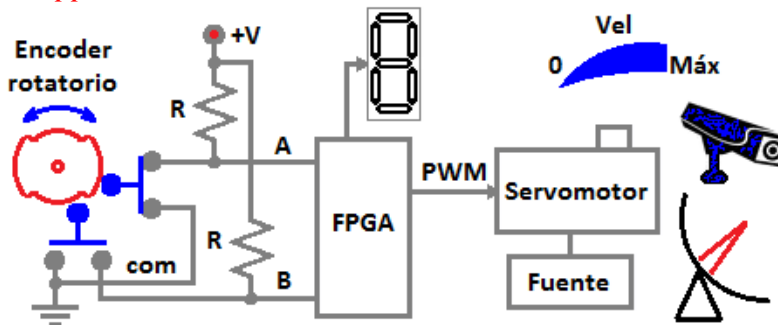


Figura 3.4. Diagrama a bloques para el control de velocidad del servo sin tope por PWM.

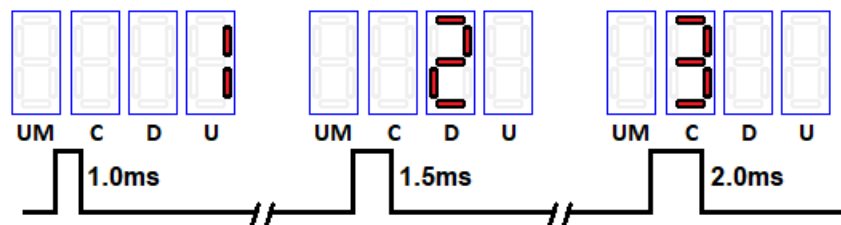


Figura 3.20. Ejemplo del valor del display para el servo con tope.

- Implementar un control de posición para un **servomotor con tope** cambiando la señal de control (PWM) adecuada al tipo de motor, por medio de un encoder rotatorio mecánico (óptico o magnético), mostrado en la figura 3.5. El movimiento del encoder en un sentido cambiará la posición del servo en ese mismo sentido, tanto horario como antihorario. Los displays (ver figura 3.21) indican el ángulo del motor (0°-180°), con resolución mínima de 5°. El motor deberá llevar una **carga** en el eje (banda, llantas, polea, sistema animatrónico, ventana o puerta a escala, etc., no se permiten cosas sencillas como hélices, clips, alambres, ligas, hojas de papel, etc.). El lenguaje en el que se implementa lo define el profesor. Reportar los códigos comentados, simulaciones, fotos del funcionamiento con texto explicativo de lo que sucede.

*Implement a position control for a servo motor with stop by changing the control signal (PWM) suitable to the motor type, by means of a mechanical rotary encoder (optical or magnetic), shown in figure 3.5. The movement of the encoder in one direction will change the position of the servo in that same direction, both clockwise and counterclockwise. The displays (see figure 3.21) indicate the angle of the motor (0°-180°), with a minimum resolution of 5°. The engine must carry a **load** on the axle (belt, wheels, pulley, animatronic system, window or door to scale, etc., simple things such as propellers, clips, wires, links, sheets of paper, etc. are not allowed). The language in which it is implemented is defined by the teacher. Report the commented codes, simulations, operation photos with explanatory text of what happens.*

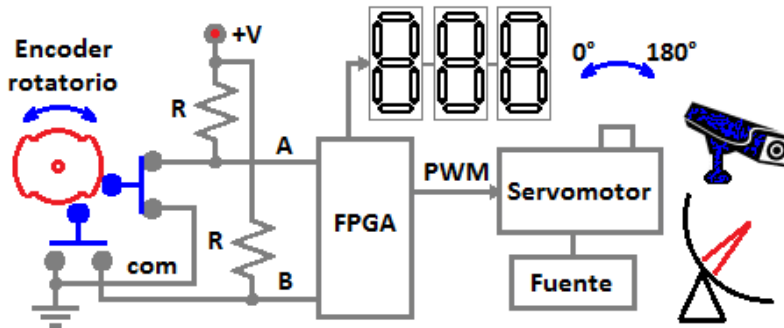


Figura 3.5. Diagrama a bloques para el control de posición del servo por PWM.

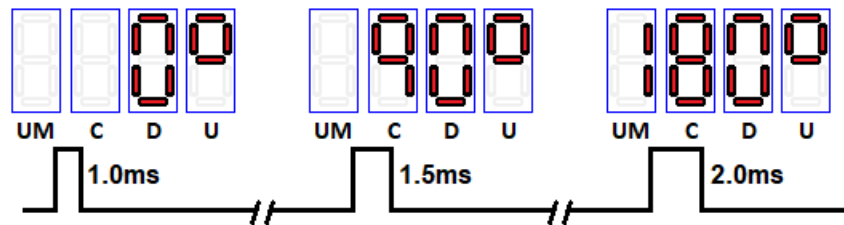


Figura 3.21. Ejemplo del valor del display para el servo con tope.

Challenge 2 (substitute the points 4 and 5).

Implement a 2DOF camera (laser, water weapon, etc.) positioner using 2 servomotors and a joystick (analog or digital). Report codes (HDL and UCF), photos and video.

- Implementar: **(a)** un codificador de teclado PS2 o USB en leds y display 7seg o LCD (mínimo 30 teclas: números, letras, símbolos y funciones) como se muestra en el diagrama a bloques de la figura 3.6. Cuando se presiona el teclado, se muestran los caracteres en el display y en código binario (scan code) en los leds. **(b)** un sistema de control de posición de dos cámaras de seguridad de 2 GDL cada una, en el que las teclas de funciones permiten elegir cualquiera de las cámaras (**F1 => cámara 1**, **F2 => cámara 2**); una vez elegida la cámara con las teclas de flechas se permite controlar los movimientos arriba-abajo-izquierda-derecha (ver figuras 3.22 y 3.23). El lenguaje en el que se implementa lo define el profesor. Reportar los códigos comentados, simulaciones, fotos del funcionamiento con texto explicativo de lo que sucede. **Nota:** Los dos incisos se revisan juntos y este punto se puede entregar hasta por dos equipos.

Implement: (a) a keyboard encoder PS2 or USB on LEDs and 7seg or LCD display (minimum 30 keys: numbers, letters, symbols and functions) as shown in the block diagram of figure 3.6. When the keyboard is pressed, the characters are shown on the display and in binary code (scan code) on the LEDs. (b) two camera security position control system of 2DOF each one, in which the function keys allow to choose any of the cameras (F1 => camera 1, F2 => camera 2); once the camera is chosen with the arrow keys, it is possible to control the movements up-down-left-right (see figures 3.22 y 3.23). The language in which it is implemented is defined by the teacher. Report the commented codes, simulations, operation photos with explanatory text of what happens. Note: the

two subsections are delivered together and this point can be delivered by up to two teams.



Figura 3.6. Diagrama a bloques para el control de posición del servo por PWM.

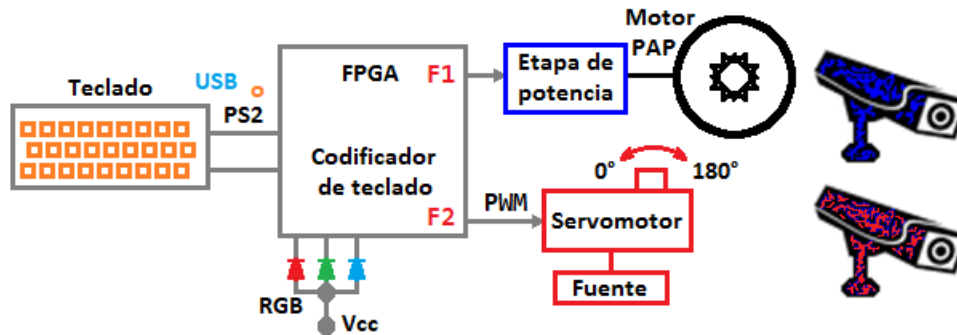


Figura 3.22. Diagrama a bloques del sistema de posición de 2 cámaras.

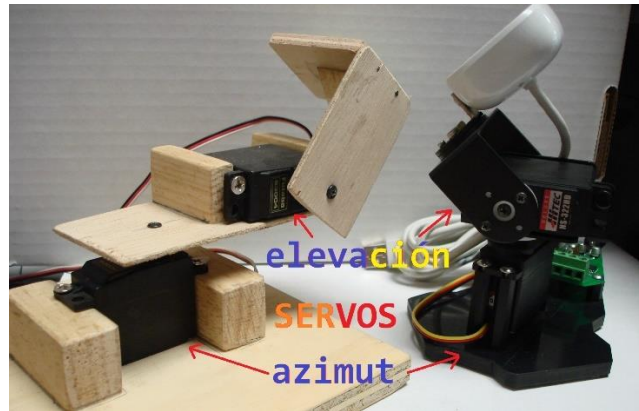


Figura 3.23. Mecanismos para el control de posición de 2 cámaras.

Challenge 3 (substitute point 6).

Implement a 2 cameras positioner controller with 2DOF each one, using 2 joysticks. Report codes (HDL and UCF), photos and video.

- Realizar sus comentarios y conclusiones (incluir en por lo menos un párrafo Gracias “Gracias a esta práctica...”). Todos los códigos van en una sola columna con comentarios editados para que no brinquen de renglón y no van en imagen.

Write your comments and conclusions. All the codes are written in one column with comments, only text, not images.