COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

INFORME TRABAJO FINAL :

MICROPROCESADORES 3ºB GITT

Jaime Martínez Martínez

Diego Luis de Mazarredo Martín

Índice

Informe Trabajo Final: Torreta Manual y Automática

Objetivo

Diseñar e implementar una torreta automática y manual controlada por un microcontrolador PIC32MX230F064D. La torreta debe ser capaz de detectar objetos mediante un sensor ultrasónico HC-SR04, ajustar su dirección utilizando un servomotor, y activar un mecanismo de disparo cuando se detecte un objetivo cercano. Además, debe permitir el control manual de la dirección mediante un joystick y la activación del disparo mediante un botón que liberará la pelota con un segundo servo.

Modos de operación

Automático**:** La torreta escanea el entorno, detecta objetos, informa al usuario a través de la UART y este decide cuando disparar.

Manual**:** El usuario controla la dirección de la torreta mediante un joystick y activa el disparo manualmente.

Diseño Hardware

Diagrama de conexión:

Componente	Módulo lógico	Puerto / Pin	Descripción
Sensor ultrasónico HC-SR04	hcsr04.c/.h	TRIG: RC4, ECHO: RC5	Detección de distancia
Motor (PWM)	motor.c/.h	PWM: OC2 → RPC8	Control de velocidad del motor
Servo principal	ModuloServo.c/.h	OC1 → RB15	Dirección de la torreta
Servo barrera	ModuloServo.c/.h	OC4 → RC6	Control de la barrera de disparo
Joystick	potenciometro.c/.h	AN0 → RA0	Control manual de dirección
Comunicación UART	uart.c/.h	TX: RB7, RX: RB13	Recepción de comandos
Botón de disparo	Main	Entrada → RB5	Activación manual del disparo
Láser	Main	Salida → RB14	Indicador visual de disparo

Diseño Software

Estructura de archivos

El software se organiza en los siguientes archivos:

1) main.c: Archivo principal que contiene la lógica del programa, incluyendo la máquina de estados y la gestión de los modos de operación.
2) hcsr04.c / hcsr04.h: Módulo que gestiona la lectura del sensor ultrasónico HC-SR04.
3) motor.c / motor.h: Módulo que gestiona el control del motor mediante PWM.
4) ModuloServo.c / ModuloServo.h: Módulo que gestiona el control de los servomotores (dirección y barrera).
5) potenciometro.c / potenciometro.h: Módulo que gestiona la lectura del joystick para el control manual.
6) uart.c / uart.h: Módulo que gestiona la comunicación UART para la recepción de comandos.
7) Retardos.c / Retardos.h: Módulo que proporciona funciones de temporización (delays).

Interacción entre módulos

La interacción entre los módulos se puede representar de la siguiente manera:

- main.c utiliza las funciones de hcsr04.c para obtener la distancia a un objeto.
- En función de la distancia, main.c ajusta la velocidad del motor mediante motor.c y la posición del servo mediante ModuloServo.c.
- En modo manual, main.c utiliza potenciometro.c para leer la posición del joystick y ajustar la dirección de la torreta.
- uart.c permite cambiar entre los modos automático y manual mediante comandos recibidos por UART. Ademas de proporcionar mensajes de información sobre la distancia detectada y la posición de la torreta.

Funciones por Módulo

hcsr04.h

- **void initHC_SR04();**
  Inicializa el sensor ultrasónico HC-SR04 configurando los pines TRIG y ECHO, y el temporizador necesario.
- **uint16_t getDistance();**
  Mide y devuelve la distancia al objeto detectado en centímetros.

motor.h

- **void initMotor();**
  Configura el PWM en el pin OC2 para controlar la velocidad del motor.
- **void controlMotor(uint16_t velocidad);**
  Ajusta la velocidad del motor en función del valor proporcionado.

ModuloServo.h

- **void inicializacionServo();**
  Inicializa los servos principal y de barrera, configurando los pines OC1 y OC4 respectivamente.
- **void sumAngulo(int delta);**
  Modifica la posición del servo principal en función del ángulo proporcionado.
- **void setGrados(int nuevo_grado);**
  Establece la posición absoluta del servo principal.
- **void activarBarrera();**
  Abre la barrera para permitir el disparo.
- **void cerrarBarrera();**
  Cierra la barrera después del disparo.
- **int estadoBarrera();**
  Devuelve el estado actual de la barrera (abierta o cerrada).

- **void delayServoPorAngulo(int delta);**
  Introduce un retardo proporcional al ángulo de movimiento del servo para asegurar un posicionamiento preciso.

potenciometro.h

- **void initADC();**
  Inicializa el ADC para leer valores analógicos del potenciómetro.
- **uint16_t leerPotenciometro();**
  Lee y devuelve el valor actual del potenciómetro.
- **int8_t adcToGrados(uint16_t adc_val);**
  Convierte el valor del ADC en un ángulo de dirección para el servo principal.

uart.h

- **void initUART();**
  Inicializa la comunicación UART configurando los pines TX y RX.
- **void sendChar(char c);**
  Envía un carácter a través de UART.
- **void sendString(const char *s);**
  Envía una cadena de caracteres a través de UART.
- **char recibirChar();**
  Recibe un carácter desde UART.
- **uint8_t UARTDisponible();**
  Verifica si hay datos disponibles para leer en UART.

Retardos.h

- **void delay_us(uint16_t us);**
  Genera un retardo en microsegundos utilizando el temporizador T5.
- **void delay_ms(uint16_t ms);**
  Genera un retardo en milisegundos utilizando la función delay_us.

Software del Microcontrolador

Estructura General del Programa (main.c)

El archivo main.c contiene la lógica principal del programa y se estructura de la siguiente manera:

1. **Inicialización de Periféricos:**
    - Configuración de UART, ADC, PWM, temporizadores y pines de entrada/salida.
    - Inicialización de módulos: sensor ultrasónico, motor, servos, potenciómetro y UART.
2. **Bucle Principal:**
    - Lectura de comandos recibidos por UART para cambiar entre modos automático y manual.
    - Ejecución de la máquina de estados correspondiente al modo actual.

Máquina de Estados

Modo Automático:

En este modo, la torreta opera de forma autónoma siguiendo una máquina de estados con los siguientes estados:

1. **Girando:**
    - El servo principal realiza un escaneo horizontal.
    - Se mide la distancia utilizando el sensor ultrasónico.
    - Si se detecta un objeto dentro de un rango predefinido, se transita al estado ESPERANDO_DISPARO.
2. **Esperando_disparo:**
    - Se espera la activación del botón de disparo.
    - Al presionar el botón, se activa la barrera y el motor para realizar el disparo.
    - Después del disparo, se transita al estado RETOMANDO.

3. **Retomando:**
    - Se cierra la barrera.
    - Se reinicia la posición del servo principal.
    - Se retorna al estado GIRANDO.

Modo Manual:

En este modo, el usuario controla directamente la torreta:

- El joystick determina la dirección del servo principal.
- El ultrasonidos determina la distancia para poder calcular la velocidad.
- El botón de disparo activa la barrera y el motor para realizar el disparo.

.

Esquema gráfico