

# Arquitectura de Computadores

## Práctica Final de laboratorio 2023-2024

**Departamento:**

Tecnología Electrónica

**Titulación:**

Grado en Informática de Gestión y Sistemas de Información

2º Curso (1º Cuatrimestre)

Octubre de 2023

José Ángel Araujo Parra  
Carlos Cuadrado Viana

Contenido

Contenido .....1

Lista de Figuras .....1

1    Introducción .....2

2    Descripción del sistema .....2

3    Modo de funcionamiento .....3

4    Temporizadores .....5

5    Secuencia detallada del software (SW) .....5

6    Programación y temporización de las tareas .....5

7    Documentación útil para el diseño .....6

8    Cambios o mejoras.....6

Lista de Figuras

F 1. Esquema electrónico del sistema de control indicado.....2

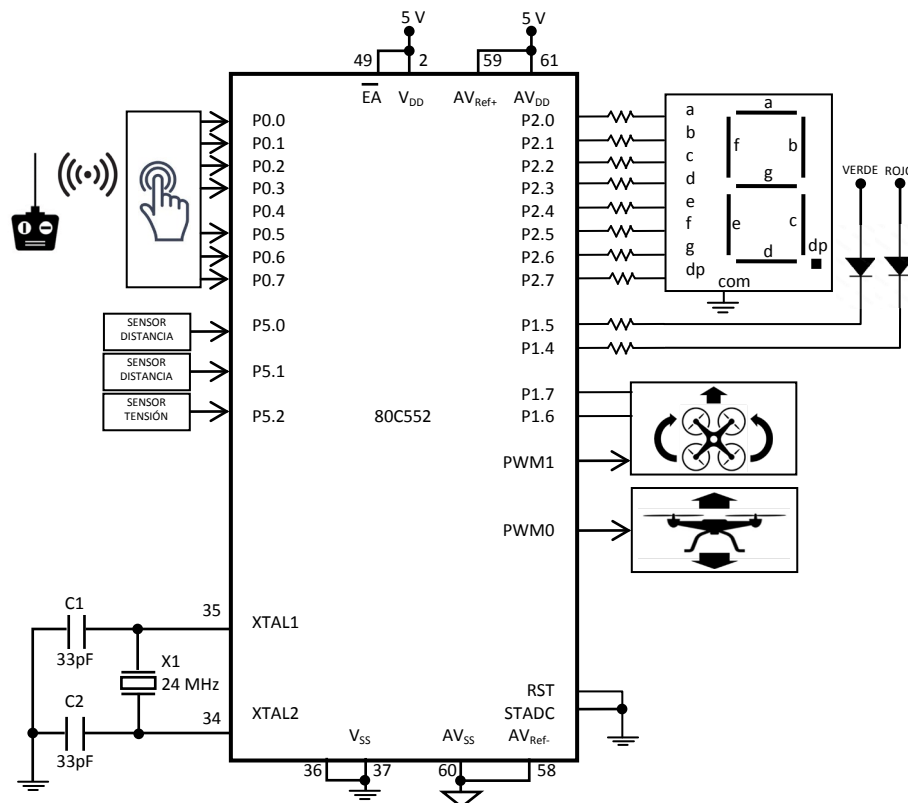
F 2. Arquitectura SW. ....5

# 1 Introducción

Este documento describe la práctica final de la asignatura de **Arquitectura de Computadores**. Esta práctica se realizará por grupos constituidos por dos personas (o una).

En esta práctica se pretende realizar el software para el controlador general de un pequeño drone ideal, dejando aparte muchas cosas que habría que tener en cuenta en la realidad y suponiendo que otras están controladas por elementos auxiliares. Por ejemplo, supondremos que está en todo momento en posición horizontal.

La figura F1 muestra el esquema hardware general del sistema de control basado en el microcontrolador 80C552.



F 1. Esquema electrónico del sistema de control indicado.

## 2 Descripción del sistema

Tendrá dos modos de funcionamiento “Automático” y “Manual”. Dispondrá de las siguientes entradas, que pueden considerarse pulsadores de un mando remoto conectados por una interface inalámbrica:

- Botón: “Despegue/Aterrizaje”. (P0.7)
- Botón: “Paro/Avanza”. (P0.6)
- Selector: “Auto/Manual”. (P0.5)
- Selectores de control manual: “g\_izda”, “g\_dcha”, “sube”, “baja”. (P0.3, P0.2, P0.1, P0.0, respectivamente)

Los dos primeros botones cambian su función alternativamente, el tercero se fija en el *drone*. Los cuatro últimos mantienen su función mientras son presionados.

- Entrada analógica del ADC que indicará la altura: "ADC\_Altura". (P5.0)
- Entrada analógica del ADC que indicará distancia de obstáculo al frente: "ADC\_Frente". (P5.1)
- Entrada analógica para medir el nivel de batería: "ADC\_Bat". (P5.2)

Por otro lado, las salidas serán las siguientes:

- PWM0: Controla la altura con el duty cycle: 0% parado, 50% mantiene altura, 70% sube y 30% baja.
- PWM1: Controla el "Giro/Avance" con el duty cycle asociado: 0% parado y 50% gira/avanza.
- Giro\_Avance: "00" 'Parado', "11" 'Avanza', "10" 'Giro\_dcha.', "01" 'Giro\_izda.'. Controla cómo se reparte el PWM1 a los rótores del drone para conseguir el giro o el avance. (P1.7 y P1.6, respectivamente)
- Led verde: Batería >10%. (P1.5)
- Led rojo: Batería <10%. (P1.4)
- Un display 7 segmentos para visualizar el nivel actual de la batería conectado al puerto 2 (P2). Entre el 0% y el 5% deberá marcar 'L' (low). Del 5% al 15%, '1'. Del 15% al 25%, '2', y así sucesivamente hasta '9'. Del 95% al 100%, debería indicar 'F' (full). El 'dp' del display indicará el modo de funcionamiento. Si se encuentra encendido, el modo actual será "Automático". Si se encuentra apagado, el modo será "Manual".

Para tener en cuenta:

- Los sensores de distancia del sistema proporcionan una salida analógica de 16,667 mV/cm, que nunca superará los 5 V. Cuando la medida es nula, la tensión de salida es 0 V.
- El sensor de tensión de la batería proporciona 4,883 mV/V (% de carga, desde 0 % al 100 %) y también tiene una referencia de 0 V cuando la batería está descargada.
- Ambas medidas de distancia deben estar completadas cada 10 ms. La cadencia de la medida de la tensión de la batería será de 10 s.
- Los drivers de motores necesitan que la orden enviada por medio de los controladores PWM tenga una frecuencia de conmutación de 500 Hz.

### 3 Modo de funcionamiento

El funcionamiento sería el siguiente:

Al encender el drone estará en reposo. Al pulsar "Despegue/Aterrizaje" el drone se pondrá en marcha y subirá hasta una altura de 1m. A partir de ahí, según esté colocado el selector en manual o automático el funcionamiento será distinto:

- **Automático:**

Directamente comenzará a avanzar:

1. Si detecta que la altura desciende por debajo de los 0,8 m activará la subida hasta alcanzar 1m.
2. Si detecta que la altura sube por encima de los 1,2 m activará la bajada hasta alcanzar 1m.
3. Si detecta un obstáculo a menos de 0,4 m se detendrá y comenzará a girar hacia la derecha hasta que la detección de obstáculo sea mayor de 0,8 m. Y

entonces, volverá a avanzar. Si después de 8 s girando (se estima que serían dos vueltas sobre sí mismo) no desaparece el obstáculo, el drone aterrizará.

4. El drone funcionará en avances de 25 segundos. Esto es, desde el momento de despegue y si no encuentra ningún obstáculo, el drone parará y girará sobre sí mismo hacia la izquierda durante 0,5 s, que supondría un giro de 45°. Si ha encontrado algún obstáculo la cuenta se reinicia.
5. Si se pulsa el botón “Despegue/Aterrizaje” o si la batería marca que queda menos del 5% el drone aterriza automáticamente: se detiene en el aire y activa el descenso, hasta que detecta una altura menor de 0,01 m, que se detiene totalmente.

- **Manual:**

En este caso funcionará el botón “Despegue/Aterrizaje” y los selectores de control manual, que indicarán las órdenes que desea ejecutar el usuario:

1. Si en el aire se pulsa el botón “Despegue/Aterrizaje” o si la batería marca que queda menos del 5% el drone aterriza automáticamente: se detiene en el aire y activa el descenso hasta que detecta una altura < 0,01 que se detiene totalmente.
2. Según estén activados los controles el drone avanzará, girará o modificará su altura, o todo a la vez. El comportamiento esperado se resume en la siguiente tabla (Tabla 1):

*Tabla 1. Comportamiento del drone en modo manual*

Avanza	Gira dcha	Gira izda	Sube	Baja	Acción
0	0	0	0	0	Parado
0	0	0	0	1	Baja
0	0	0	1	0	Sube
0	0	0	1	1	-
0	0	1	0	0	Gira izda
0	0	1	0	1	Gira izda + baja
0	0	1	1	0	Gira izda + sube
0	0	1	1	1	-
0	1	0	0	0	Gira dcha
0	1	0	0	1	Gira dcha + baja
0	1	0	1	0	Gira dcha + sube
0	1	0	1	1	-
0	1	1	0	0	-
0	1	1	0	1	-
0	1	1	1	0	-
0	1	1	1	1	-
1	0	0	0	0	Avanza
1	0	0	0	1	Baja + avanza
1	0	0	1	0	Sube + avanza
1	0	0	1	1	-
1	0	1	0	0	Gira izda
1	0	1	0	1	Gira izda + baja
1	0	1	1	0	Gira izda + sube
1	0	1	1	1	-
1	1	0	0	0	Gira dcha
1	1	0	0	1	Gira dcha + baja
1	1	0	1	0	Gira dcha + sube
1	1	0	1	1	-
1	1	1	0	0	-
1	1	1	0	1	-
1	1	1	1	0	-
1	1	1	1	1	-

## 4 Temporizadores

El microprocesador **80C552** dispone de 3 timers, **T0**, **T1** y **T2**. T0 y T1 poseen la misma funcionalidad. T2 tiene funcionalidades adicionales.

Para implementar las temporizaciones que requiere el diseño SW, se pueden usar cualquiera de estos temporizadores (uno o varios). En cuanto a la señal de reloj a usar en estos temporizadores, se es libre de usar bien el reloj del microcontrolador de 24 MHz o un reloj externo (ver figura 1). En este último caso, se deberá de indicar la frecuencia de reloj externo que se ha elegido.

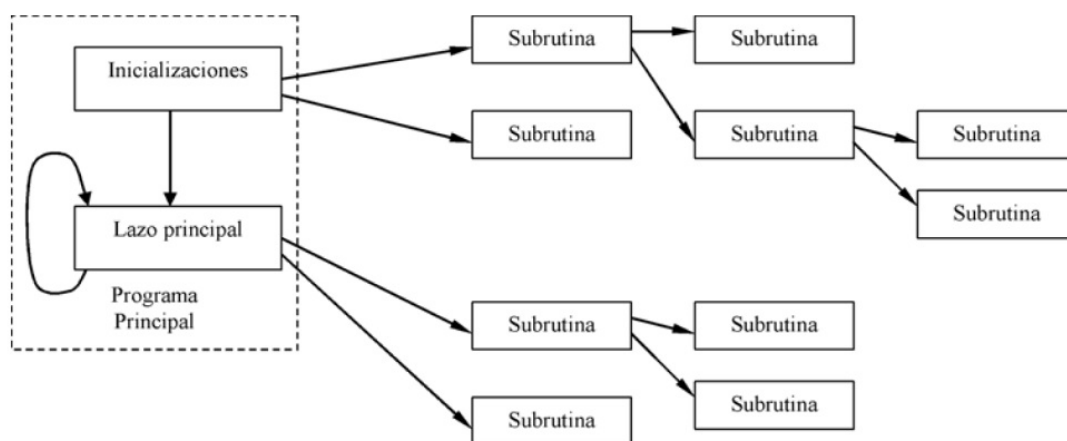
- El funcionamiento de los timers T0 y T1 se encuentra descrito en las páginas 8 a 10 del documento **doc1\_80C51\_family\_program\_guide**.
- El funcionamiento del timer T2 se encuentra descrito en las páginas 6 a 9 del documento **doc2\_80C51\_family\_derivatives**.

## 5 Secuencia detallada del software (SW)

La estructura del software de control que se pide diseñar será algo similar a lo explicado en clase, que se reproduce en la figura F 2.

En el lazo principal, y eventualmente en la inicialización, se usarán subrutinas para realizar los procesos de tipo repetitivo. Una puede ser la subrutina que actualiza lo que se muestra por los displays, otra puede ser la que reprograma el PWM0... y otras más.

En el momento de encendido del drone, el programa hará un proceso de inicialización para programar puertos y periféricos de la manera adecuada para el uso que se les va a dar. Acto seguido entraremos en el lazo principal.



F 2. Arquitectura SW.

## 6 Programación y temporización de las tareas

El programa implementado en el microcontrolador tiene que ser capaz de cumplir con lo especificado en los apartados anteriores y tiene que consistir obligatoriamente en una máquina de estados-eventos-acciones.

Las acciones a realizar por el programa se darán como respuesta a la activación de cualquiera de los eventos.

El procedimiento para desarrollar el programa comenzará desarrollando el diagrama de estados-eventos-acciones. Una vez éste esté terminado se procederá a definir los diagramas de flujo. Éstos tienen que llegar a definir el programa principal, la máquina de estados y las máquinas de eventos para cada estado. Llegado a este punto se procederá a la programación de las acciones correspondientes, definiendo previamente los correspondientes diagramas de flujo. Algunos de estos diagramas serán pedidos en la memoria del proyecto. Todos estos diagramas serán presentados al profesor y si éste da su visto bueno se procederá a codificar el programa en el entorno de desarrollo Keil.

## 7 Documentación útil para el diseño

- **doc1\_80C51\_family\_program\_guide.pdf:** En este documento se pueden consultar todo lo relativo al juego de instrucciones del  $\mu$ C, además de la configuración y programación del timer 0 para la gestión de eventos.
- **doc2\_80C51\_family\_derivatives.pdf:** Este documento muestra las diferencias entre el  $\mu$ C 8051 y el 80C552. La configuración y programación del convertidor analógico digital, así como la programación de las interrupciones y la gestión del módulo PWM se encuentra descrito en este documento.
- **doc3\_80C552\_chip\_datasheet.pdf:** En este documento provee información sobre el conector y pines disponibles en el microprocesador y puede ser de ayuda a la hora de entender el esquema eléctrico del proyecto.
- **Macro Assembler and Utilities for 8051 and Variants:** Este documento viene con la instalación de la herramienta Keil y se encuentra en **C:\Keil\C51\HLP** (esto puede variar con la instalación), teniendo como nombre **A51.pdf**.

## 8 Cambios o mejoras

Una vez terminado el diseño, se podrán realizar mejoras o cambios, que darían una aportación adicional al diseño. Antes de realizar las mejoras, estas deben ser planteadas al profesor. Las aportaciones tendrán un reflejo positivo en la evaluación del proyecto.