Tema 1

Arquitectura ESP32 y Bloques de Radiocomunicación

ESP32 - Fundamentos de Radiocomunicaciones

Duración: 20 horas | **Evaluación:** 1ª Evaluación

Resultados de Aprendizaje y Criterios de Evaluación

RA 1: Determina los bloques constructivos de los equipos de radiocomunicaciones, reconociendo sus módulos y componentes y midiendo parámetros.

Criterios de Evaluación:

- **CE a)** Se ha identificado la función de los módulos que componen los equipos de radiocomunicaciones (audiofrecuencia, osciladores y frecuencia intermedia, entre otros). **0,2 puntos**
- **CE b)** Se han diferenciado las señales de modulación de amplitud y frecuencia de los equipos analógicos. **0,2 puntos**
- **CE c)** Se han especificado las señales de transmisión digital. **0,2 puntos**
- **CE d)** Se han medido los parámetros fundamentales de los equipos y módulos. **0,2 puntos**
- **CE e)** Se han comparado las señales de entrada y salida de los módulos con las indicadas en el manual técnico. **0,2 puntos**
- **CE f)** Se han relacionado las medidas obtenidas con las características de los módulos. **0,2 puntos**

Puntuación Total RA 1: 1,2 puntos

1.1 Arquitectura del ESP32

1.1.1 Características Generales

El ESP32 es un microcontrolador de 32 bits desarrollado por Espressif Systems, diseñado específicamente para aplicaciones IoT y de conectividad inalámbrica.

Características principales:

- **CPU Dual-Core:** Dos procesadores Xtensa LX6 a 240 MHz
- **Memoria:** 520 KB SRAM, 4 MB Flash (configurable)
- **Conectividad:** WiFi 802.11 b/g/n, Bluetooth 4.2 y BLE
- **GPIO:** 34 pines programables
- **Periféricos:** ADC, DAC, PWM, SPI, I2C, UART, I2S
- **Seguridad:** Hardware de encriptación AES, SHA-2, RSA

1.1.2 Bloques Funcionales de Radiocomunicación

El ESP32 integra varios bloques especializados para radiocomunicaciones:

Bloque WiFi:

- **RF Frontend:** Amplificador de bajo ruido (LNA), mezclador, filtros
- **Baseband Processor:** Procesamiento de señales digitales
- **MAC Layer:** Control de acceso al medio
- **Protocol Stack:** Implementación de protocolos WiFi

Bloque Bluetooth:

- **RF Transceiver:** Transmisor y receptor de 2.4 GHz
- **Baseband Controller:** Procesamiento de señales Bluetooth
- **Link Manager:** Gestión de conexiones
- **HCI Interface:** Interfaz de control del host

1.2 Configuración del Entorno de Desarrollo

1.2.1 Arduino IDE

Configuración del entorno de desarrollo para ESP32:

```
// Configuración básica del ESP32 void setup() { // Inicializar comunicación serie Serial.begin(115200); // Configurar GPIO pinMode(2, OUTPUT); // LED integrado // Configurar WiFi WiFi.mode(WIFI_STA); Serial.println("ESP32 inicializado correctamente"); } void loop() { // Parpadeo del LED digitalWrite(2, HIGH); delay(1000); digitalWrite(2, LOW); delay(1000); }
```

1.2.2 ESP-IDF (Espressif IoT Development Framework)

Framework oficial de Espressif para desarrollo avanzado:

```
// Ejemplo básico en ESP-IDF #include "freertos/FreeRTOS.h" #include "freertos/task.h" #include "driver/gpio.h" #define LED_PIN GPIO_NUM_2 void app_main() { // Configurar GPIO gpio_config_t io_conf = {}; io_conf.intr_type = GPIO_INTR_DISABLE; io_conf.mode = GPIO_MODE_OUTPUT; io_conf.pin_bit_mask = (1ULL << LED_PIN); io_conf.pull_down_en = 0; io_conf.pull_up_en = 0; gpio_config(&io_conf); // Bucle principal while(1) { gpio_set_level(LED_PIN, 1); vTaskDelay(1000 / portTICK_PERIOD_MS); gpio_set_level(LED_PIN, 0); vTaskDelay(1000 / portTICK_PERIOD_MS); } }
```

1.3 Periféricos de Comunicación

1.3.1 GPIO (General Purpose Input/Output)

El ESP32 cuenta con 34 pines GPIO configurables:

```
// Configuración de GPIO #define LED_PIN 2 #define
BUTTON_PIN 0 void setup() { pinMode(LED_PIN, OUTPUT);
pinMode(BUTTON_PIN, INPUT_PULLUP); } void loop() { if
(digitalRead(BUTTON_PIN) == LOW) {
digitalWrite(LED_PIN, HIGH); } else {
digitalWrite(LED_PIN, LOW); } }
```

1.3.2 ADC (Analog-to-Digital Converter)

Conversión de señales analógicas a digitales:

```
// Lectura de ADC #define ADC_PIN A0 void setup() {
Serial.begin(115200); analogReadResolution(12); // 12
bits de resolución } void loop() { int adcValue =
analogRead(ADC_PIN); float voltage = (adcValue * 3.3) /
4095.0; Serial.printf("ADC: %d, Voltaje: %.2f V\n",
adcValue, voltage); delay(1000); }
```

1.3.3 DAC (Digital-to-Analog Converter)

Generación de señales analógicas:

```
// Generación de señal analógica #define DAC_PIN DAC1
void setup() { // Configurar DAC
dac_output_enable(DAC_PIN); } void loop() { // Generar
onda senoidal for (int i = 0; i < 360; i++) { float
angle = i * PI / 180.0; int value = (int)(127.5 + 127.5
```

```
* sin(angle)); dac_output_voltage(DAC_PIN, value);
delay(10); } }
```

1.4 Módulos de Entrada de Audiofrecuencia

1.4.1 Micrófonos Digitales

Interfaz con micrófonos digitales I2S:

```
#include "driver/i2s.h" #define I2S_WS 25 #define
I2S_SD 33 #define I2S_SCK 32 void setup() {
i2s_config_t i2s_config = { .mode = (i2s_mode_t)
(I2S_MODE_MASTER | I2S_MODE_RX), .sample_rate = 44100,
.bits_per_sample = I2S_BITS_PER_SAMPLE_16BIT,
.channel_format = I2S_CHANNEL_FMT_ONLY_LEFT,
.communication_format = I2S_COMM_FORMAT_I2S,
.intr_alloc_flags = ESP_INTR_FLAG_LEVEL1,
.dma_buf_count = 8, .dma_buf_len = 64 };
i2s_pin_config_t pin_config = { .bck_io_num = I2S_SCK,
.ws_io_num = I2S_WS, .data_out_num = I2S_PIN_NO_CHANGE,
.data_in_num = I2S_SD }; i2s_driver_install(I2S_NUM_0,
&i2s_config, 0, NULL); i2s_set_pin(I2S_NUM_0,
&pin_config); }
```

1.4.2 ADC de Audio

Captura de señales de audio analógicas:

```
// Captura de audio con ADC #define AUDIO_PIN A0 void
setup() { Serial.begin(115200);
analogReadResolution(12);
analogSetAttenuation(ADC_11db); } void loop() { int
audioSample = analogRead(AUDIO_PIN); // Convertir a
```

```
voltaje float voltage = (audioSample * 3.3) / 4095.0;
// Enviar por serie Serial.println(voltage); delay(1);
// 1ms = 1kHz de muestreo }
```

1.5 Osciladores y Generadores de Frecuencia

1.5.1 Cristales y Osciladores

El ESP32 utiliza cristales de 40 MHz para el sistema principal y 32.768 kHz para el RTC.

Características del oscilador principal:

- **Frecuencia:** 40 MHz
- **Estabilidad:** ±20 ppm
- **Temperatura:** -40°C a +85°C
- **Envejecimiento:** ±5 ppm/año

1.5.2 PLL (Phase-Locked Loop)

El ESP32 incluye PLLs para generar frecuencias de reloj:

```
// Configuración de frecuencia de CPU #include
"esp_system.h" void setup() { // Configurar frecuencia
de CPU setCpuFrequencyMhz(240); // Máxima frecuencia
Serial.begin(115200); Serial.printf("Frecuencia CPU: %d
MHz\n", getCpuFrequencyMhz());
Serial.printf("Frecuencia APB: %d MHz\n",
getApbFrequency() / 1000000); }
```

1.6 Amplificadores de RF

1.6.1 LNA (Low Noise Amplifier)

Amplificadores de bajo ruido integrados en el ESP32:

Características del LNA:

- **Ganancia:** 20-30 dB
- **Ruido:** < 2 dB
- **Frecuencia:** 2.4 GHz
- **Consumo:** 20-30 mA

1.6.2 Etapas de Potencia

Amplificadores de potencia para transmisión:

```
// Configuración de potencia de transmisión WiFi
#include "esp_wifi.h" void setup() {
WiFi.mode(WIFI_STA); // Configurar potencia de
transmisión esp_wifi_set_max_tx_power(84); // 20 dBm
máximo // Conectar a red WiFi.begin("SSID",
"password"); }
```

Materiales Necesarios

🔧 ESP32 DevKit V1 o V2 (1 unidad por grupo)

🔧 Protoboard (1 por grupo)

🔧 Cables de conexión (jumper wires)

🔧 LEDs y resistencias (220Ω)

- 🔧 Pulsadores
- 🔧 Potenciómetros (10kΩ)
- 🔧 Micrófono digital I2S (opcional)
- 🔧 Osciloscopio (1 por laboratorio)
- 🔧 Multímetro (1 por grupo)
- 🔧 Generador de señales (1 por laboratorio)

Actividades Teóricas

- 📋 **Actividad 1.1:** Investigar las características técnicas del ESP32 y compararlas con otros microcontroladores
- 📋 **Actividad 1.2:** Analizar el diagrama de bloques del ESP32 e identificar los módulos de radiocomunicación
- 📋 **Actividad 1.3:** Estudiar los diferentes tipos de modulación digital (ASK, FSK, PSK)
- 📋 **Actividad 1.4:** Investigar las especificaciones de los osciladores y cristales utilizados en el ESP32
- 📋 **Actividad 1.5:** Analizar las características de los amplificadores de RF integrados

Prácticas de Laboratorio

Práctica 1.1: Configuración Básica del ESP32

Objetivo: Familiarizarse con el entorno de desarrollo y realizar las primeras configuraciones.

Duración: 2 horas

Entregables: Código fuente comentado, mediciones de consumo, informe técnico

Práctica 1.2: Configuración de GPIO y Periféricos

Objetivo: Configurar y utilizar los periféricos básicos del ESP32.

Duración: 3 horas

Entregables: Esquema de conexiones, código fuente, mediciones de señales

Práctica 1.3: Interfaz de Audio Digital

Objetivo: Implementar la captura de audio usando I2S.

Duración: 3 horas

Entregables: Código I2S, análisis de señales, informe de rendimiento

Práctica 1.4: Medición de Parámetros RF

Objetivo: Medir y analizar los parámetros de los módulos de radiofrecuencia.

Duración: 4 horas

Entregables: Mediciones de potencia, análisis de espectro, informe técnico

Criterios de Evaluación

Instrumentos de evaluación:

- **Examen teórico (30%):** Conceptos de arquitectura ESP32, bloques de radiocomunicación
- **Prácticas de laboratorio (50%):** Implementación, mediciones, documentación
- **Trabajo de investigación (20%):** Análisis de especificaciones técnicas

Entregables obligatorios:

- Código fuente de todas las prácticas
- Informes técnicos con mediciones
- Esquemas de conexiones
- Análisis de resultados

Recursos Adicionales

- **Documentación oficial:** ESP-IDF Programming Guide
- **Arduino Core:** ESP32 Arduino Core
- **Datasheet ESP32:** Especificaciones técnicas completas
- **Herramientas de medición:** Manuales de osciloscopio y analizador de espectro

---

IES Universidad Laboral de Albacete | Departamento de Electrónica | Curso 2025-2026

Profesor: Carlos Víllora Fernández | **Tema 1:** Arquitectura ESP32 y Bloques de Radiocomunicación