

Universidad Autónoma de Baja California
Facultad de Ciencias Químicas e Ingeniería



MICROCONTROLADORES

Practica No. 2

Configuración del Ambiente de Desarrollo

Docente: Garcia Lopez Jesus Adan

Alumno: Gómez Cárdenas Emmanuel Alberto

Matricula: 01261509

Objetivo:

El alumno instalará el ambiente de desarrollo y se familiarizará con las herramientas.

Material:

- Computadora Personal

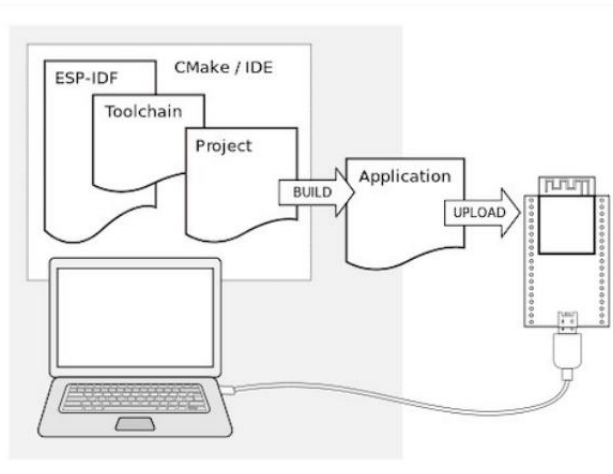
Teoría:

- IDEs para sistemas embebidos (pros y contras).
- Técnicas de depuración para sistemas embebidos.
- Investigar aplicaciones de Terminal serial (VT100)

Desarrollo

1. Instalación de VS Code.
2. Instalación de la extensión de ESP-IDF dentro de VS Code.
3. Abrir la extensión de ESP-IDF y realizar la instalación del SDK. Pueden apoyarse en la Guía Oficial.
4. Leer la guía de Inicio del ESP-IDF: <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/index.html>
5. Familiarizarse con la tarjeta de desarrollo que han adquirido.
6. Utilizar el archivo semilla del repositorio y compilar para verificar que las herramientas se han instalado correctamente. Dentro de este ejemplo podrán ver como se podría usar printf() al incluir (y el SDK se encarga de conectar la tubería). Para poder ver las impresiones, es necesario el uso de una terminal.
7. Cargar el binario generado a la tarjeta de desarrollo, en caso de no contar con el HW, apoyarse con el simulador en línea Wokwi (<https://wokwi.com/projects/new/esp32>).

Verificar la información del microcontrolador en la impresión del programa de ejemplo. Deberían de ver algo como: "This is esp32 chip with 2 CPU core(s), WiFi/BT/BLE, silicon revision 0, 4MB external flash" A partir de este punto ya tenemos todos los requisitos básicos para el ambiente de desarrollo:



Actividades a realizar:

1. Realizar los cambios necesarios para que ahora parpadee en código morse su nombre.

```
/*
 * Autor: Emmanuel Alberto Gómez Cárdenas
 * Fecha: 21/02/2023
 * Descripcion: Practica 1, Configuracion de ambiente de desarrollo
 * Laboratorio de la clase Microcontroladores
 */

#include <stdio.h>
#include "sdkconfig.h"
#include "freertos/FreeRTOS.h"
#include "freertos/task.h"
#include "esp_system.h"
#include "esp_spi_flash.h"
#include "driver/gpio.h"

#define LED_GPIO GPIO_NUM_2
#define DOT 300
#define DASH (DOT * 3)
#define PAUSE 1
#define BREAK (DOT * 5)
#define END 0

uint16_t mA[] = {DOT, DASH, END};
uint16_t mE[] = {DOT, END};
uint16_t mL[] = {DOT, DASH, DOT, DOT, END};
uint16_t mM[] = {DASH, DASH, END};
uint16_t mN[] = {DASH, DOT, END};
uint16_t mU[] = {DOT, DOT, DASH, END};
uint16_t mEND[] = {END};
uint16_t *name[] = {mE, mM, mM, mA, mN, mU, mE, mL, mEND};

void app_main() {
    gpio_reset_pin(LED_GPIO);
    gpio_set_direction(LED_GPIO, GPIO_MODE_OUTPUT);
    printf("Hello world!\n");

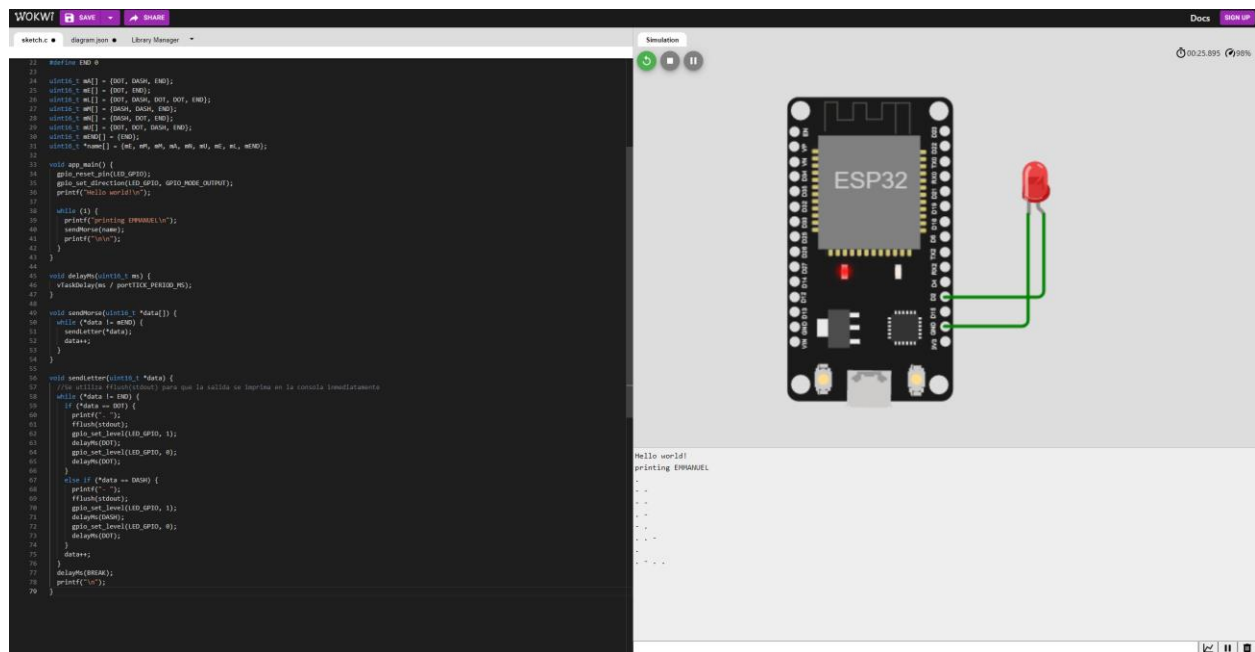
    while (1){
        printf("printing EMMANUEL\n");
        sendMorse(name);
        printf("\n\n");
    }
}
```

```
void delayMs(uint16_t ms) { vTaskDelay(ms / portTICK_PERIOD_MS); }

void sendMorse(uint16_t *data[]) {
    while (*data != mEND) {
        sendLetter(*data);
        data++;
    }
}

void sendLetter(uint16_t *data) {
    //Se utiliza fflush(stdout) para que la salida se imprima en la consola
    inmediatamente
    while (*data != END) {
        if (*data == DOT) {
            printf(". ");
            fflush(stdout);
            gpio_set_level(LED_GPIO, 1);
            delayMs(DOT);
            gpio_set_level(LED_GPIO, 0);
            delayMs(DOT);
        }
        else if (*data == DASH) {
            printf("- ");
            fflush(stdout);
            gpio_set_level(LED_GPIO, 1);
            delayMs(DASH);
            gpio_set_level(LED_GPIO, 0);
            delayMs(DOT);
        }
        data++;
    }
    delayMs(BREAK);
    printf("\n");
}
```

2. Anexar captura del analizador lógico como verificación.



Conclusiones y comentarios

El ambiente de desarrollo ESP-IDF ofrece una plataforma de desarrollo eficiente para desarrollar proyectos utilizando los microcontroladores ESP32, así mismo ESP-IDF en VS Code ofrece una experiencia familiar y agradable para aquellas personas acostumbradas a trabajar en VS Code.

Debido a que aún no cuento con el microcontrolador en esta práctica he utilizado me he apoyado con el simulador para realizar los ejercicios propuestos.

Bibliografía

Wikimedia Foundation. (2023, February 7). *ESP32*. Wikipedia. Retrieved February 21, 2023, from <https://en.wikipedia.org/wiki/ESP32>

Get started. ESP. (n.d.). Retrieved February 21, 2023, from <https://docs.espressif.com/projects/esp-idf/en/stable/esp32/get-started/index.html>