**Marco Teórico**

En este apartado se determinan los conocimientos teóricos necesarios para la comprensión de la construcción del Sistema de control de aforos en instituciones educativas basada en el internet de las cosas.

**Sistema embebido**

Un sistema embebido está diseñado para realizar funciones específicas y sus componentes se encuentran integrados en una placa, cuyo procesamiento central se lleva a cabo gracias a un microcontrolador.

Los sistemas embebidos pueden ser programados directamente en el lenguaje ensamblador del microcontrolador o usando lenguajes como C o C++ mediante compiladores específicos. También se suelen programar con ayuda de los kits de desarrollo, elementos de hardware que facilita las pruebas, la programación y el prototipado.

Los microcontroladores integrados en el mismo chip, como las placas de desarrollo compatibles con Arduino, facilitan el lidiar con problemas que tengan los diferentes métodos de flasheado y compilación del microcontrolador.

**Microcontrolador**

Un microcontrolador es un circuito integrado que es el componente principal de una aplicación embebida. Es como si fuera una pequeña computadora que incluye elementos de entrada y salida. También incluye un procesador (CPU) y memoria donde puede almacenar el programa y sus variables (flash y RAM). Su principal función es automatizar procesos y procesar la información.

Un microcontrolador debe de tener al menos

* Microprocesador
* Periféricos (Unidades de Entrada/Salida)
* Memoria

Diagrama

Descripción generada automáticamente

Figura 0. Esquema general de un Microcontrolador.

Existen diversos fabricantes de microcontroladores tales como Altera, Atmel, Intel, Silicon Motion, Texas Instruments, Toshiba, Tensilica, etc. Para este proyecto se utiliza el microcontrolador Xtensa Dual-Core LX6 de 32 bits fabricado por Tensilica, el cual podemos encontrar en las placas de desarrollo ESP32



Figura 0. Microcontrolador Xtensa Dual-Core LX6

**Placa de desarrollo ESP32**

ESP32 es una familia de microcontroladores, es una serie de SoC (por sus siglas en inglés, System on Chip) y módulos de bajo costo y bajo consumo de energía creado por la empresa Espressif Systems.

**Imagen que contiene electrónica, circuito

Descripción generada automáticamente**

Figura 0. ejemplos de placas ESP32.

En esencia, estos se basan en un microprocesador Tensilica Xtensa LX6 (de uno o dos núcleos) con una frecuencia de operación de hasta 240 MHz.

**ESP32 DEVKIT DOIT**

En este proyecto se utilizó la placa de desarrollo ESP32 DEVKIT V1 de 30 Pines, para la construcción del circuito.



Figura 0. ESP32 DEVKIT V1 DOI

El ESP32 DEVKIT V1 es un microcontrolador de bajo costo y consumo de energía, cuenta con tecnología Wi-Fi y Bluetooth de modo dual integrada que permite controlar todo tipo de sensores, módulos y actuadores: es el sucesor del microcontrolador ESP8266**.**

Te permite generar proyectos de Internet de las cosas “IoT” de forma eficiente y económica, ya que integra internamente una gran cantidad de periféricos incluyendo: sensores táctiles capacitivos, sensor de efecto Hall, amplificadores de bajo ruido, interfaz para tarjeta SD, Ethernet, SPI de alta velocidad, UART, I2S e I2C.

|  |  |
| --- | --- |
| Especificaciones – ESP32 DEVKIT V1 DOIT | |
| Número de núcleos | 2 (doble núcleo) |
| Wi-Fi | 2,4 GHz hasta 150 Mbits/s |
| Bluetooth | BLE (Bluetooth Low Energy) y Bluetooth heredado |
| Arquitectura | 32 bits |
| Frecuencia de reloj | Hasta 240 MHz |
| CARNERO | 512 KB |
| Pins | 30 o 36 (depende del modelo) |
| Periféricos | Táctil capacitiva, ADC (convertidor analógico a digital), DAC (convertidor digital a analógico), I2C (circuito interintegrado), UART (receptor/transmisor asíncrono universal), CAN 2.0 (Netwokr de área del controlador), SPI (interfaz periférica serie), I2S (sonido inter-IC integrado), RMII (interfaz independiente de medios reducido), PWM (modulación de ancho de pulso) y más. |

Tabla 0. Especificaciones – ESP32 DEVKIT V1 DOI.

**Guía de pines ESP32 DEVKIT DOIT.**

El chip ESP32 viene con 48 pines con múltiples funciones. No todos los pines están expuestos en todas las placas de desarrollo ESP32, y algunos pines no se pueden usar.

Los periféricos ESP32 incluyen:

* 18 canales de convertidor de analógico a digital (ADC)
* 3 interfaces SPI
* 3 interfaces UART
* 2 interfaces I2C
* 16 canales de salida PWM
* 2 convertidores de digital a analógico (DAC)
* 2 interfaces I2S
* 10 GPIO de detección capacitiva

Las características ADC (convertidor analógico a digital) y DAC (convertidor digital a analógico) se asignan a pines estáticos específicos. Sin embargo, puede decidir qué pines son UART, I2C, SPI, PWM, etc., solo necesita asignarlos en el código. Esto es posible gracias a la función de multiplexación del chip ESP32.

Aunque puede definir las propiedades de los pines en el software, hay pines asignados de forma predeterminada como se muestra en la siguiente figura

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 0. GPIOs – ESP32 DEVKIT V1 DOI Versión 30 Pines.

**Métodos de comunicación - ESP32.**

**Puerto Serial (UART).**

Un puerto serial es un módulo de comunicación digital para sistemas embebidos, permite la comunicación entre dos dispositivos digitales. Cuenta con dos conexiones RX y TX, estos nos indican los modos de comunicación que pueden manejar:

Simplex: transmisión de datos en una dirección.

Half-duplex: transmisión de datos en cualquier dirección, pero no simultáneamente.

Full-duplex: transmisión de datos en ambas direcciones simultáneamente.

La transmisión de UART convierte los datos en paralelo del dispositivo maestro (por ejemplo, la CPU) en forma de serie y los transmite en serie a la UART receptora. Después volverá a convertir los datos en serie en datos paralelos para el dispositivo receptor.

Su funcionamiento es, primero que quiere transmitir datos, recibirá datos de un bus de datos que es enviado por otro componente (por ejemplo, la CPU). Después de obtener los datos del bus de datos, agregará un bit de inicio, un bit de paridad y un bit de parada para crear el paquete de datos. Este paquete de datos se transmite luego por el pin TX donde el UART receptor leerá el paquete de datos en su pin RX. Los datos se envían hasta que no quedan datos en el UART transmisor, y se recibe el bit de parada.

Por tanto, cada paquete UART estará compuesto de los siguientes bits: 1 bit de inicio (nivel bajo), 8 bits de datos, incluido el bit de paridad, 1 o 2 bits de parada (nivel alto).

Imagen que contiene objeto, reloj, dibujo, medidor

Descripción generada automáticamente

Figura 0. Ejemplo de comunicación UART.

**Protocolo I2C.**

I2C (Inter Integrated Circuit) es un puerto y protocolo de comunicación sincrónico, multi-maestro, multi-esclavo, el puerto I2C es uno de los más utilizados en aplicaciones embebidas. Define la trama de datos y las conexiones físicas para transferir bits entre 2 dispositivos digitales. Este protocolo permite conectar hasta 127 dispositivos esclavos a la vez en el mismo bus.

Este protocolo utiliza dos líneas para controlar otros dispositivos. La primera línea es el reloj llamado SCL (línea de reloj en serie), mientras que la otra es para datos llamado SDA (puerto de aceptación de línea de datos). Este protocolo es similar al UART, sin embargo, no se utiliza para la comunicación PC-dispositivo, sino con módulos y sensores, siendo el PC el “maestro” y el resto los llamados “esclavos”.

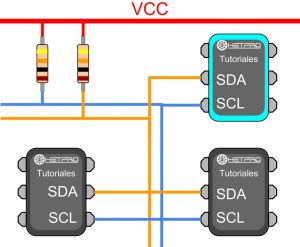


Figura 0. **Conexión de tres dispositivos a un bus de comunicación I2C.**

**Protocolo SPI.**

SPI (Serial Peripheral Interface) o interfaz de periféricos en serie, siendo un protocolo similar al I2C, pero especialmente pensado para microcontroladores. También es usado en situaciones donde la velocidad es importante como, por ejemplo, tarjetas SD, módulos de visualización o cuando la información se actualiza y cambia rápidamente como termómetros.

Este protocolo de comunicación serial es de tipo serie de tipo síncrono y consta de dos líneas de datos (MOSI y MISO), una línea de reloj (SCK) y una línea de selección esclava (SS).

* MOSI: Master Output Slave In - Salida del maestro entrada del esclavo (línea a través de la cual el maestro envía datos a sus esclavos)
* MISO - Master In Slave Out – Entrada del maestro salida del esclavo (línea a través de la cual los esclavos responde al maestro)
* SCK - Serial Clock (reloj proporcionado por el dispositivo maestro)
* SS - Slave Select (línea utilizada para seleccionar el esclavo con el que el maestro desea comunicarse)

Por lo general es un protocolo más rápido que el I2C debido a que es más simple. Sin embargo, existen límites prácticos cuando se requiere de múltiples esclavos, ya que cada esclavo requiere una señal de habilitación separada lo que ocasiona más complejidad a medida que aumentan los esclavos.

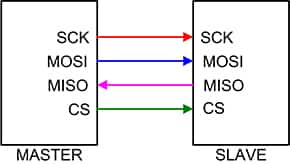


Figura 0. **Conexión SPI Mastes - Slave.**

|  |  |  |  |
| --- | --- | --- | --- |
| Protocolo | UART | I2C | SPI |
| Complejidad | Fácil | Fácil al conectar múltiples dispositivos | Difícil a medida que aumentan los dispositivos |
| Velocidad | Es más lento | Mas rápido que UART | Es más rápido |
| Numero de Dispositivos | Permite hasta 2 | Permite hasta 127, pero se vuelve complejo | Permite muchos, pero se vuelve complejo |
| Duplex | Full-duplex | Half-duplex | Full-duplex |

Tabla 0. Comparación de los métodos de comunicación ESP32.

**Entornos de programación ESP32.**

El ESP32 se puede programar en diferentes entornos de programación. Puede utilizar: Arduino IDE, Espressif IDF, Micropython, JavaScript LUA, etc.

En este proyecto, se programa el ESP32 con Arduino IDE.

**Arduino IDE.**

El entorno de desarrollo integrado (IDE) de Arduino es una aplicación multiplataforma (Windows

macOS, Linux) que está desarrollada en el lenguaje de programación Java. Se utiliza para escribir y

cargar programas en placas compatibles con Arduino, pero también, con la ayuda de núcleos de

terceros, se puede usar con placas de desarrollo de otros proveedores.

El IDE de Arduino admite los lenguajes C y C ++ utilizando reglas especiales de estructuración de códigos.​ El IDE de Arduino suministra una biblioteca de software del proyecto Wiring, que proporciona muchos procedimientos comunes de E/S.

En pocas palabras Arduino es una plataforma de creación de electrónica de código abierto, la cual está basada en hardware y software libre, flexible y fácil de utilizar para los creadores y desarrolladores. Esta plataforma permite crear diferentes tipos de microordenadores de una sola placa a los que la comunidad de creadores puede darles diferentes tipos de uso

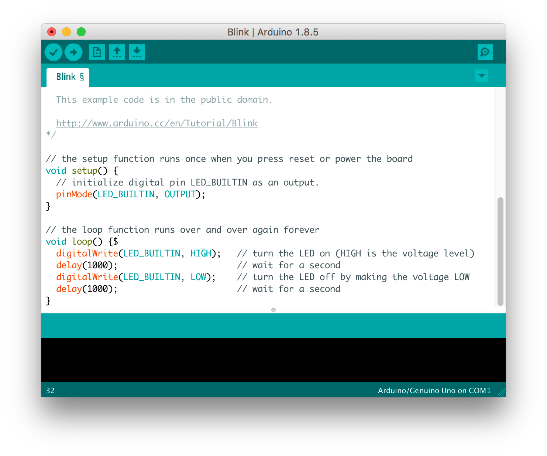


Figura 0. IDE Arduino.