TALLER DE SISTEMAS EMBEBIDOS

SISTEMA DE CONTROL DE ESTACIONAMIENTO

SALAMUNICH MIRKO - 110958

ACOSTA MATEO - 109391

TORREALBA MIGUEL - 107309

CABALLERO ORLANDO - 104013

PROBLEMATICA

Sin un sistema de control en un estacionamiento:

- Confusion y quejas por desconocer la tarifa previo al ingreso
- Mayor probabilidad de errores de cobro
- Demoras al ingresar al estacionamiento
- Ingresos innecesarios al no haber lugar disponible
- Maniobras peligrosas en busca de espacios para estacionar
- Perdida de clientes y de ingresos economicos

NECESIDAD

Para resolver la problematica planteada se necesita un sistema que

- Muestre la cantidad de lugares disponibles dentro del estacionamiento
- Realice el calculo de tarifa por tiempo de estadia

OBJETIVOS

• Brindar informacion clara y precisa en tiempo real sobre tarifas y lugares disponibles

• Optimizar el uso del espacio y el flujo de circulacion

•Evitar congestiones en los ingresos o egresos

• Agilizar ingresos y salidas

• Garantizar cobros precisos y transparentes

• Mejorar la experiencia y satisfaccion del usuario

BENEFICIOS PARA LOS USUARIOS

- Saben si hay lugar disponible en el estacionamiento sin perder tiempo
- Evitan vueltas innecesarias buscando espacio dentro del estacionamiento
- Reciben una tarifa justa y transparente
- Mejora en la rapidez y la comodidad de la experiencia

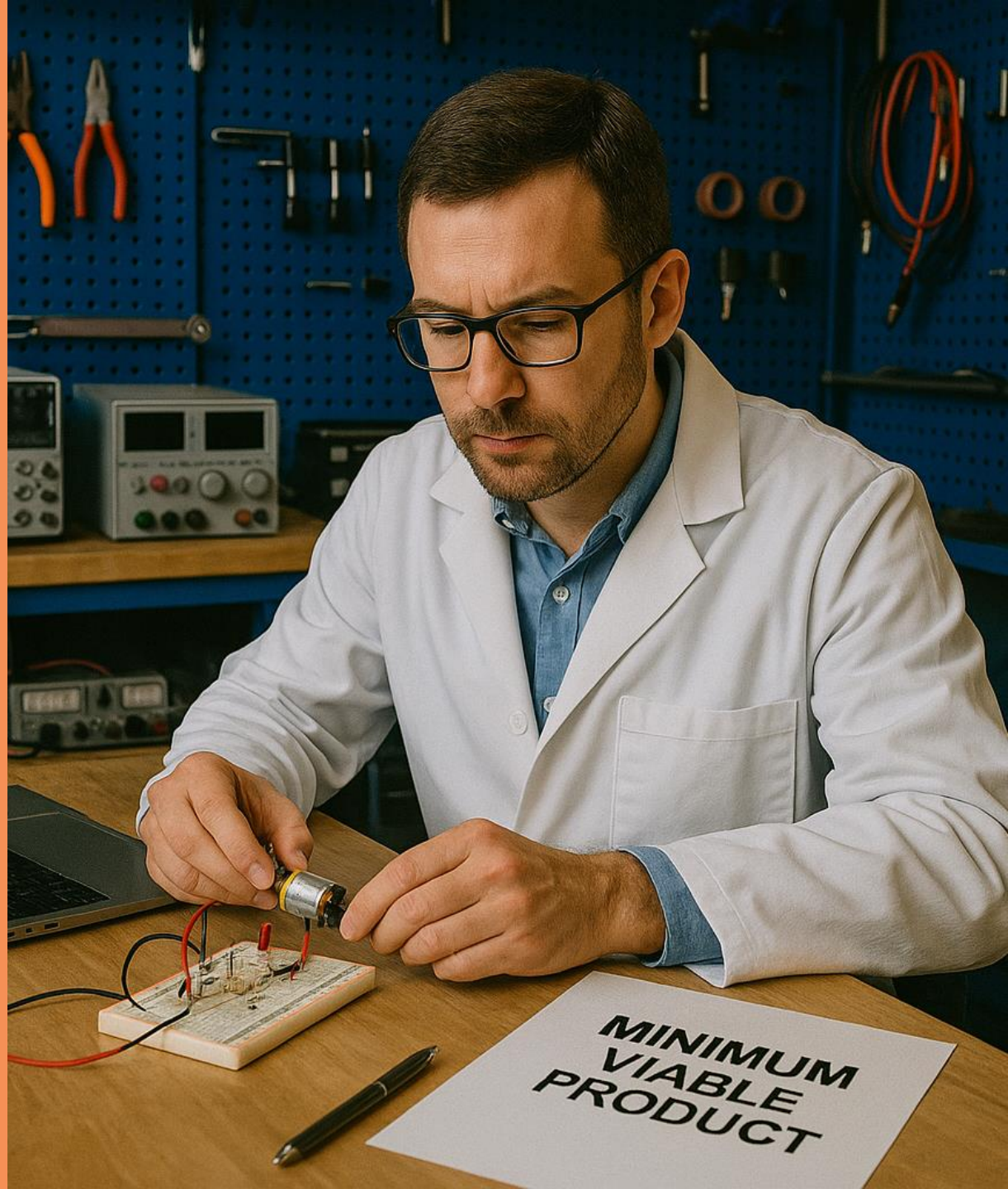
BENEFICIOS PARA LOS ADMINISTRADORES

- Controlan en tiempo real la ocupacion del estacionamiento
- Optimizacion del uso de los espacios
- Reduccion de quejas y conflictos con los clientes
- Mayor eficiencia en el cobro y gestion de ingresos
- Mejora en la imagen debido a la actualizacion tecnologica
- Mejora en la competitividad del servicio frente a otros

SISTEMA EMBEBIDO

Se define a un sistema embebido como aquel sistema electrónico programable diseñado para realizar funciones específicas dentro de un dispositivo, integrando hardware y software.

- Automatiza procesos y reduce intervencion humana
- Aumenta la velocidad de operacion y la precision
- Permite el control el tiempo real
- Mejora la eficiencia energetica y funcional



PRODUCTO MINIMO VIABLE

Es la versión más básica de un producto que incluye solo las funciones esenciales. el fin de validar su utilidad antes de invertir en un desarrollo completo.

- Permite probar la idea rápidamente con menor costo.
- Facilita recibir retroalimentación real de los usuarios.
- Reduce riesgos antes de producir en gran escala.
- Ayuda a enfocar el desarrollo en lo que realmente aporta valor.

¿QUE FUNCIONES VA A CUMPLIR EL DISPOSITIVO?

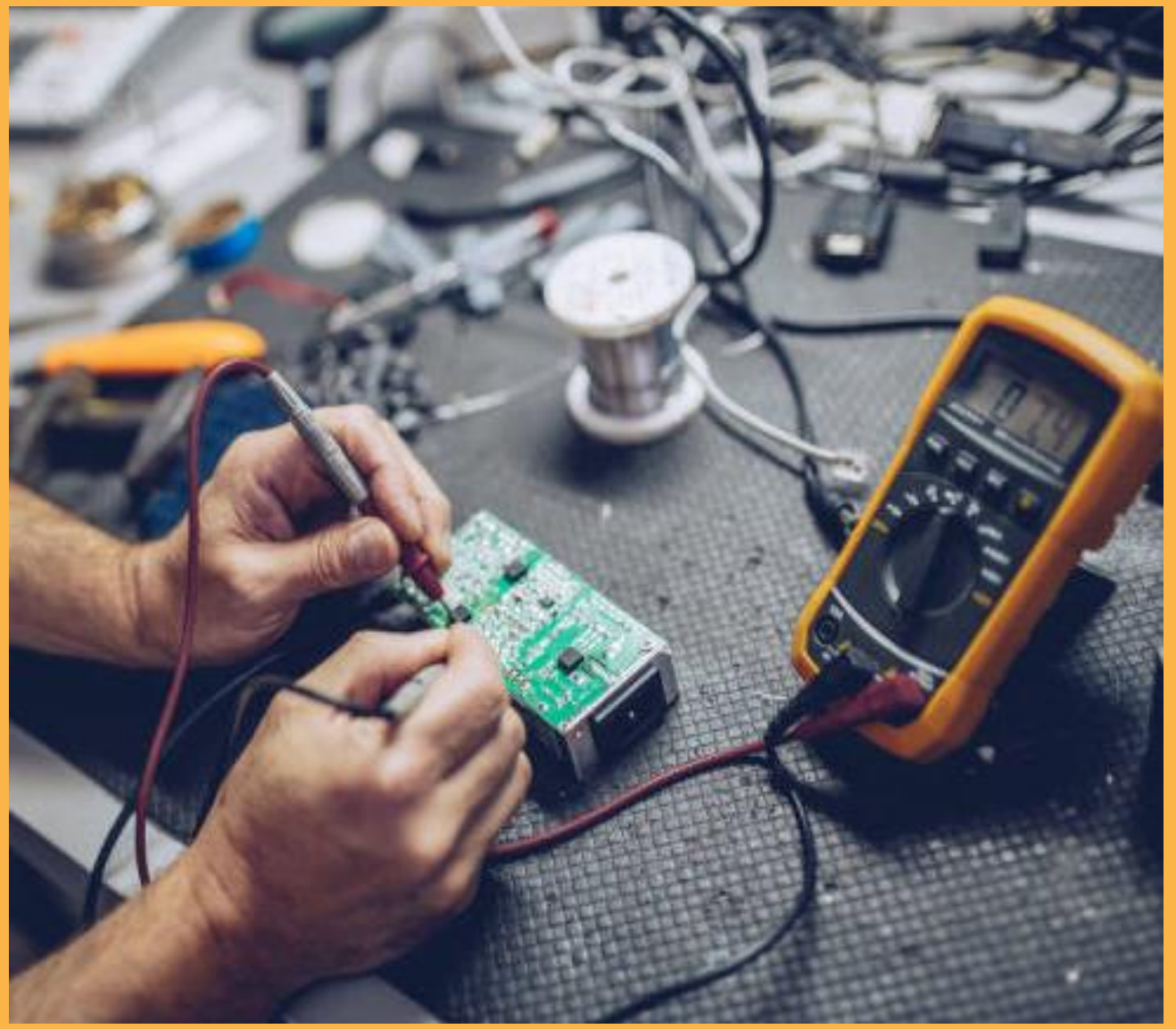
- Calcular el tiempo transcurrido desde el ingreso hasta el egreso del vehiculo
- Modo NORMAL para que el usuario interactue con el sistema y deposite su informacion
- Modo SETUP para la configuracion de tarifas por tipo de vehiculo
- Registrar la hora y la patente en la que ingresa un usuario a dejar su vehiculo
- Almacenar informacion en una memoria externa arduino
- Mostrar la informacion por pantalla al usuario

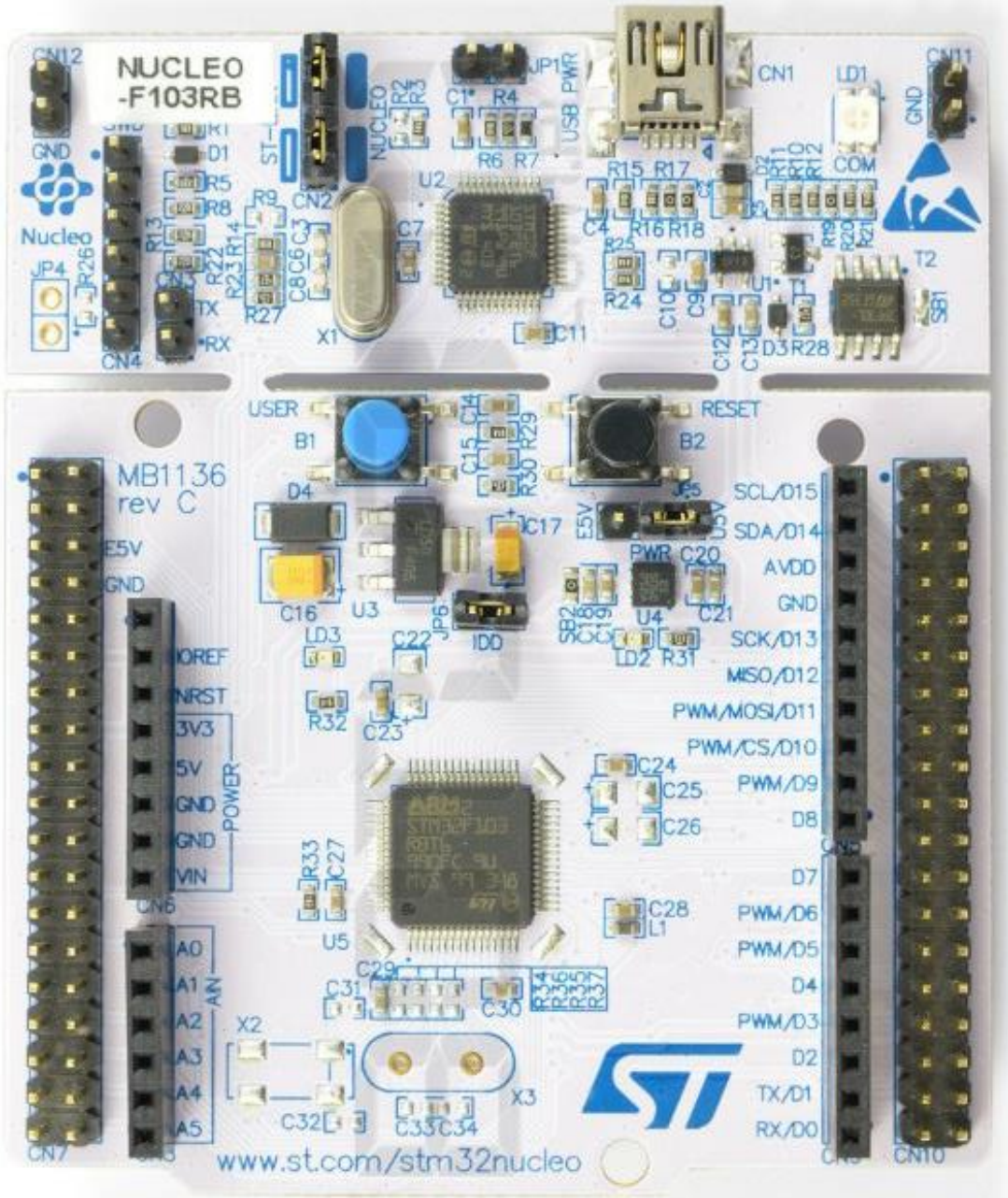
ESTRUCTURA DEL FUNCIONAMIENTO

- Tipos de vehiculos que registrara el dispositivo: motos, autos, camiones
- El usuario ingresa mediante un teclado matricial los ultimos tres numeros de su patente
- El precio de la tarifa se informa por pantalla cuando el cliente egresa, y depende del tiempo de estadia y el tipo de vehiculo registrado.

¿QUE COMPONENTES SE NECESITAN?

- Placa STM-32-Nucleo-F103RB
- Modulo de memoria EEPROM
- Display LCD
- Teclado matricial





PLACA STM-32-NUCLEO-F103RB

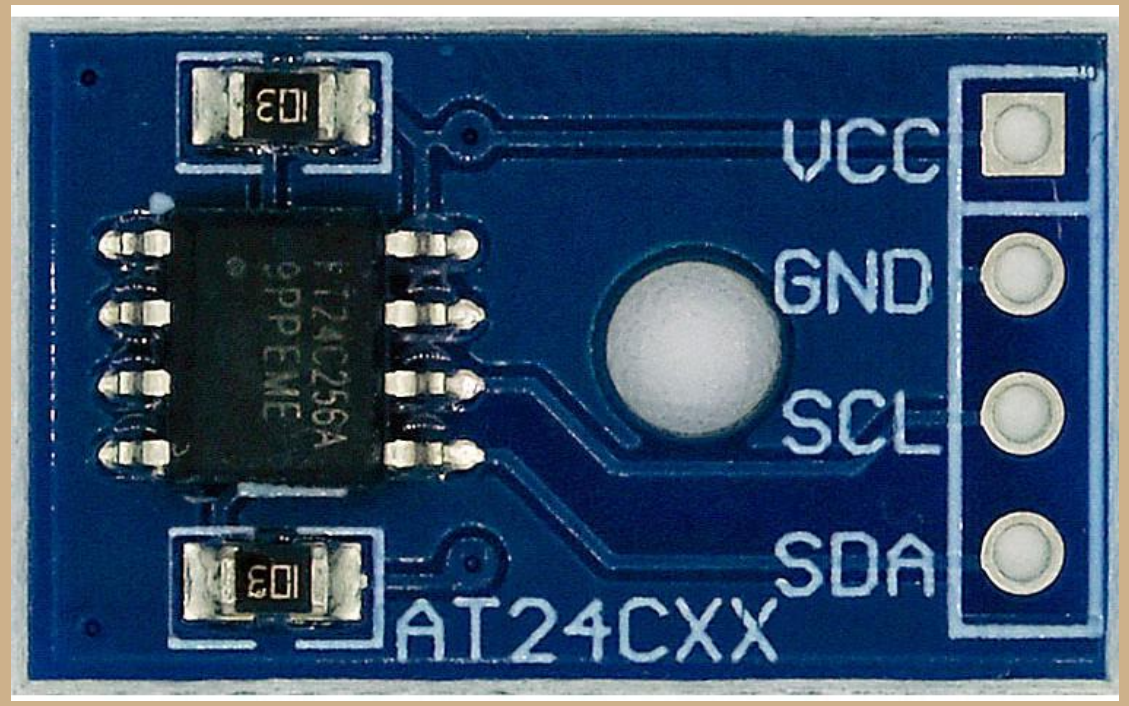
- Control de perifericos de entrada y salida, digitales y analogicos
- Ejecucion de programas en tiempo real
- Soporta conexiones USB para la programacion y manipulacion de datos
- Incluye timers para medir tiempos y generar señales

TECLADO MATRICIAL

¿Que es?
Es unconjunto de botones organizados enfilas y columnas para ingresar datos

¿Para que sirve?

- Ingresar numeros y datos en un sistema
- Controlar dispositivos mediante combinaciones de teclas
- Facilitar la interaccion con microcontroladores y placas



MODULO DE MEMORIA EEPROM

¿Que es?

Es un conjunto fisico que contiene una memoria EEPROM y almacena datos permanentemente en un sistema electrico

¿Para que sirve?

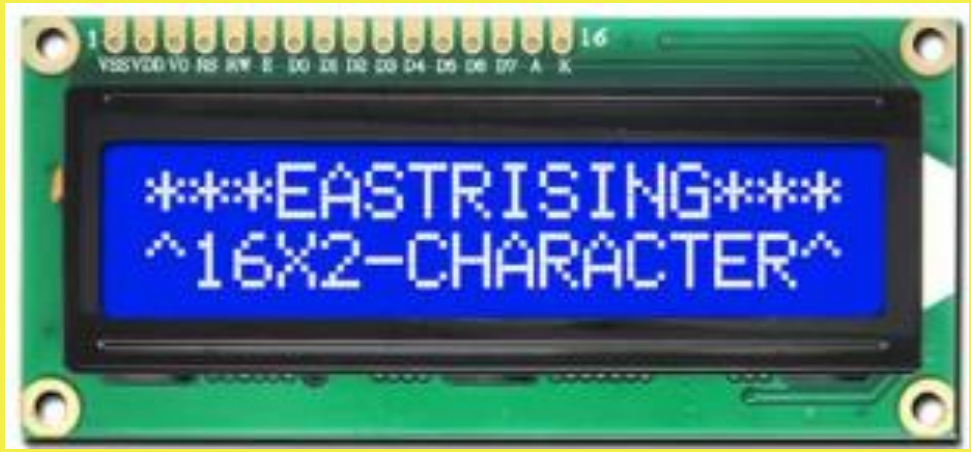
- Guardar instrucciones basicas del sistema
- Almacenar datos
- Proteger informacion contra modificaciones
- Garantizar el arranque del dispositivo

DISPLAY LCD

¿Que es?
Es una pantalla que muestra información usando cristal líquido.

¿Para que sirve?

- Mostrar datos al usuario
- Visualizar mensajes o instrucciones
- Visualizar el estado de un dispositivo
- Mostrar resultados de mediciones
- Facilitar la interaccion con sistemas electronicos



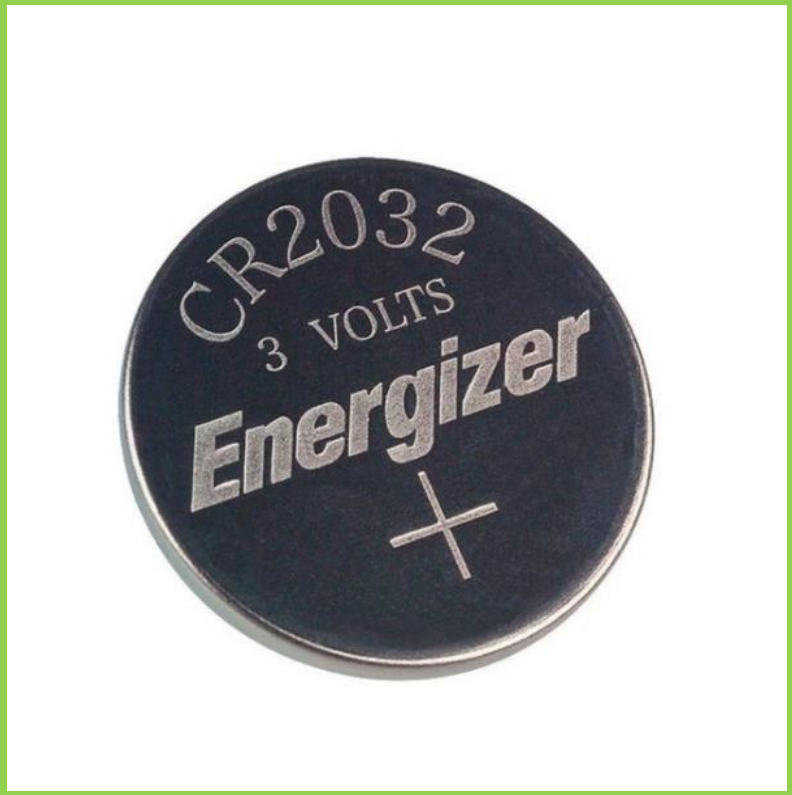
PILA DE LITIO

Caracteristicas

- Bateria de 3 V
- Compacta y de larga duracion
- Ideal para dispositivos de pequeño porte

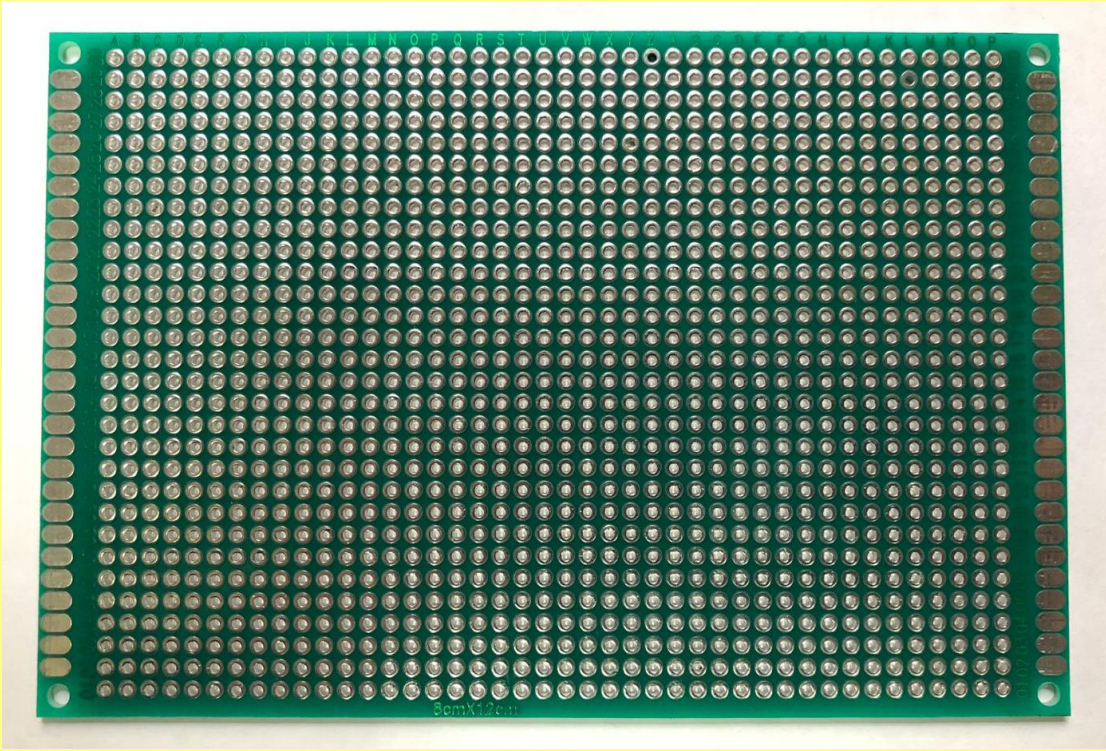
Funciones a cumplir

- Alimenta el reloj interno del microcontrolador RTC de manera constante aunque no este conectado

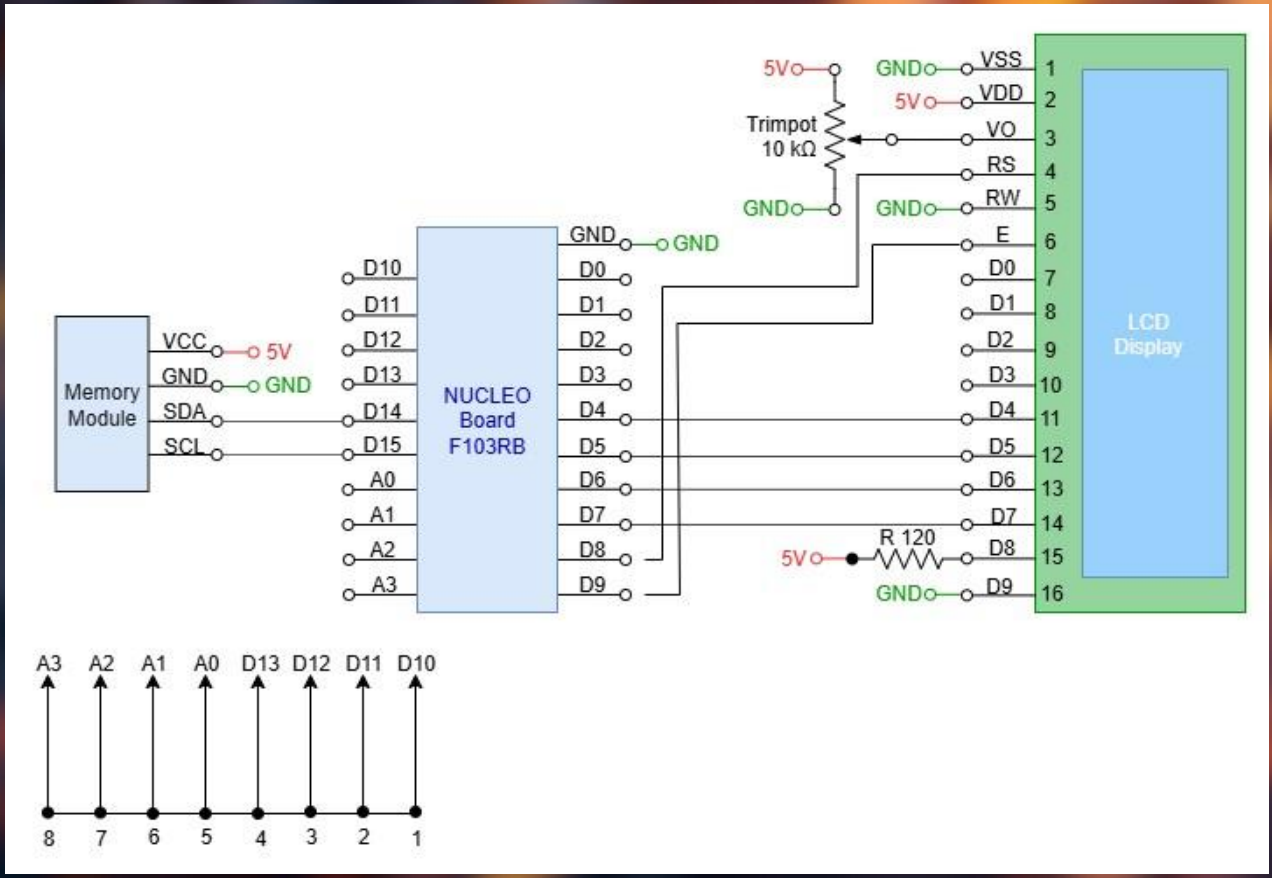


PLACA DE DESARROLLO

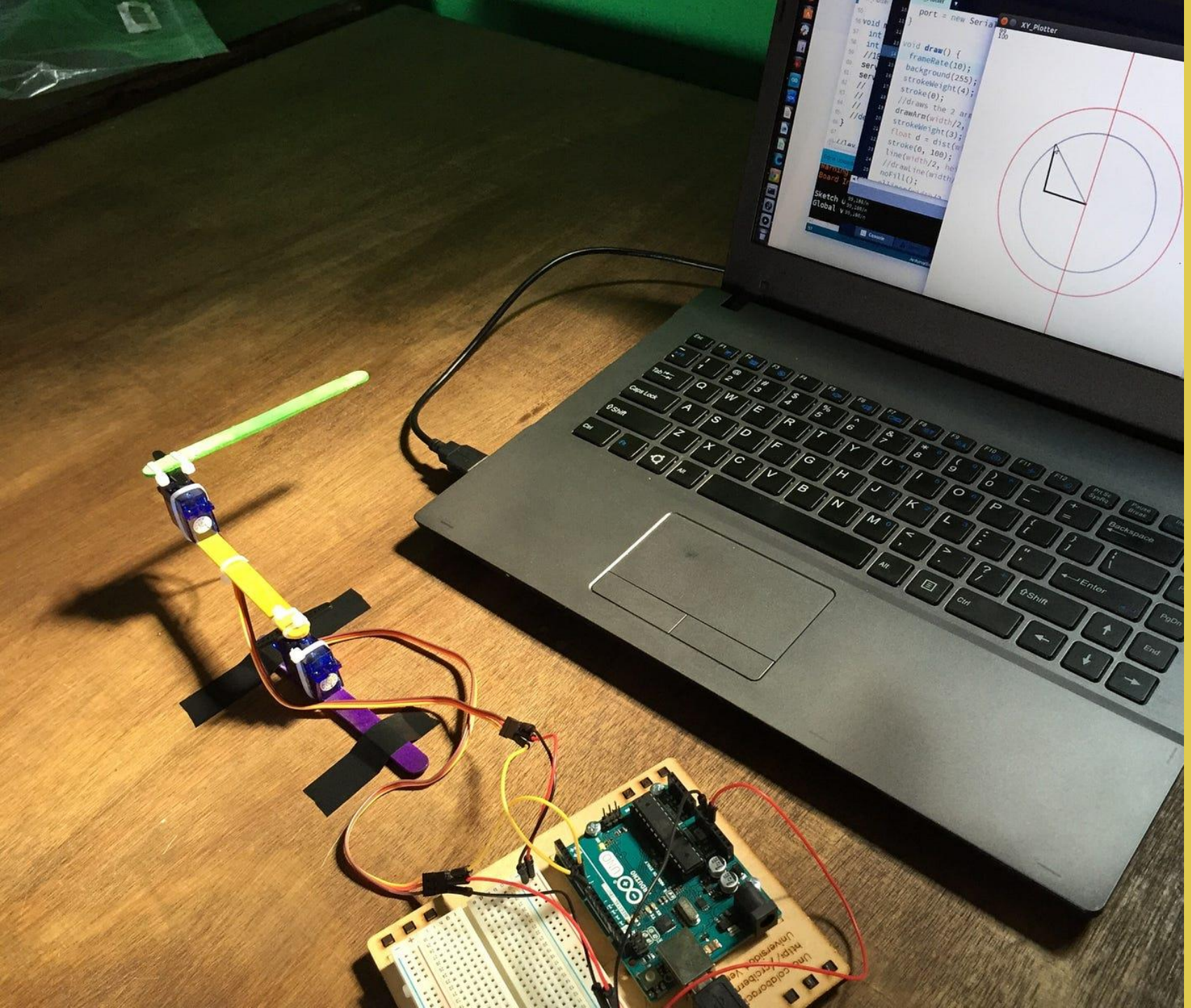
- Montar y soldar componentes electronicos
- Conectar circuitos de forma permanente
- Dar soporte fisico y electrico al proyecto



CONEXIONES



- La placa Nucleo F103RB funciona como el cerebro del sistema. Recibe y manda datos de la memoria. Controla el Display LCD
- El Display LCD recibe datos y comandos de la placa Nucleo F103RB. Esta conectado en formato de 8 bits. Se alimenta con 5 V y GND, y con un trimpot se ajusta el contraste de texto.
- El módulo de memoria se conecta a la placa Nucleo F103RB a través del bus I²C utilizando las líneas SDA y SCL. Recibe alimentación de 5 V y GND.



¿QUE HERRAMIENTAS SE NECESITAN?

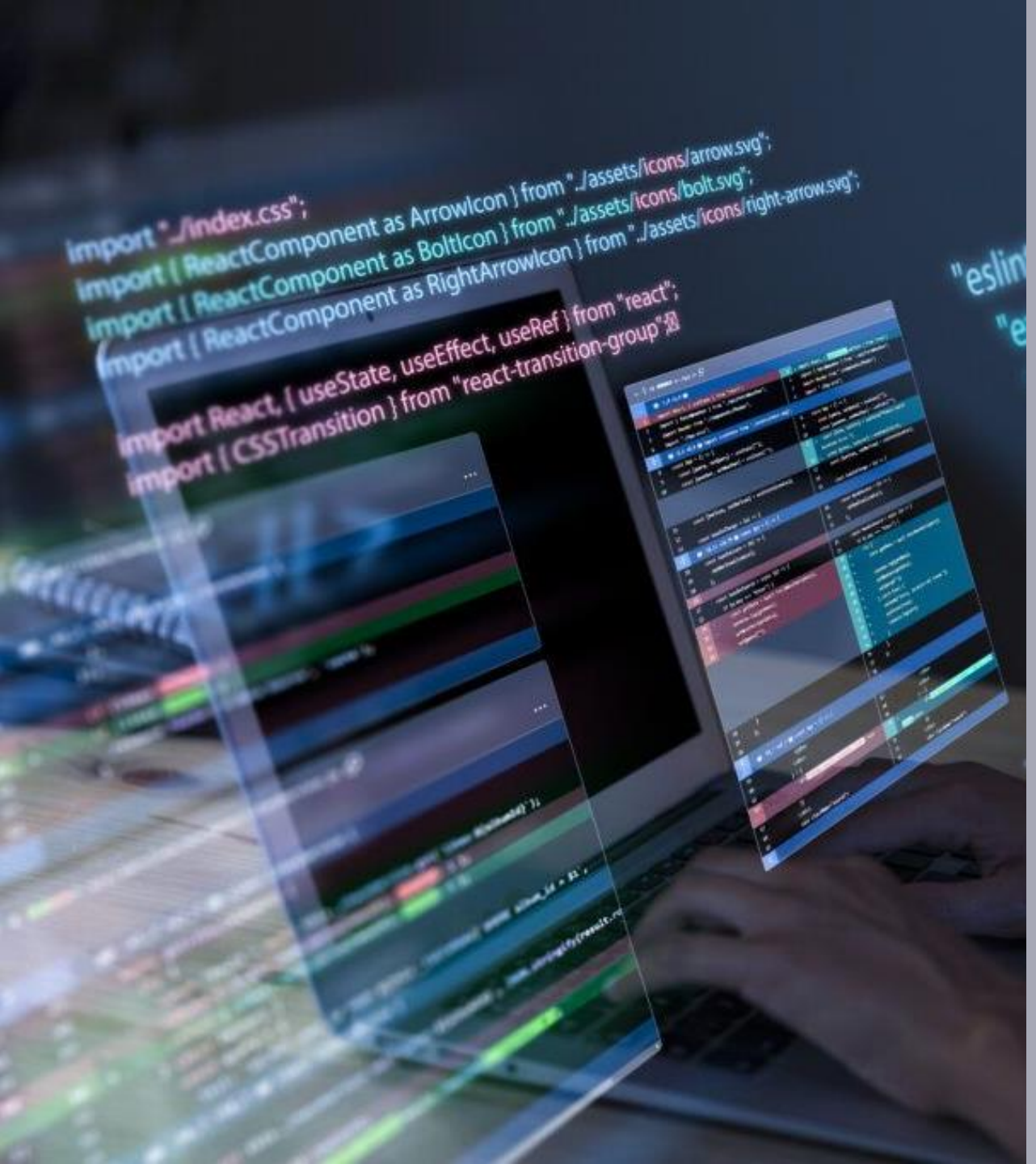
Para determinar la estructura y el funcionamiento del dispositivo se necesitan herramientas para depurar software y diagramas de estado

ITEMIS CREATE

¿Que es?
Es un programa para diseñar diagramas de estados y lógicas de control de manera visual.

¿Para que sirve?

- Crear diagramas de estados facilmente
- Organizar transiciones y acciones de un sistema
- Simular el comportamiento antes de programar
- Ahorrar tiempo en el diseño de la logica compleja
- Documentar el funcionamiento de un proyecto claramente



STM-32-CUBE-IDE

¿Que es?

Es un programa que permite desarrollar, configurar y probar aplicaciones para placas STM32.

¿Para que sirve?

- Permite programar y compilar un codigo para que la placa lo ejecute
- Facilitacion para configurar perifericos
- Configura el refloj interno para ajustar velocidad y rendimiento
- Cuenta con herramientas de depuracion para encontrar errores
- Mantiene actualizado el firmware pack de la placa

DIAGRAMA DE ESTADO

- Modularidad: Permite dividir el sistema en modulos, facilitando la comprension
- Estados: Representan las condiciones o modos de operacion del sistema
- Subestados: Detallan el comportamiento interno dentro de un estado principal
- Transiciones: Indican los cambios de estados que mejoran la trazabilidad del flujo del sistema

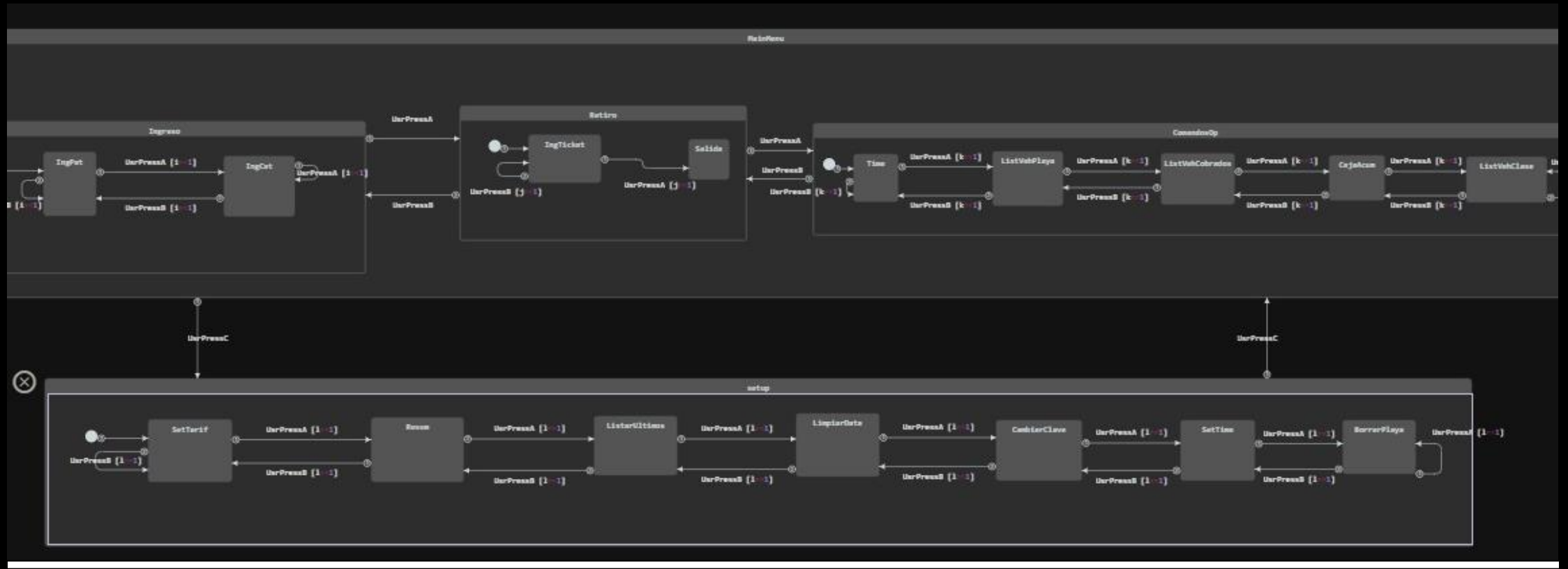
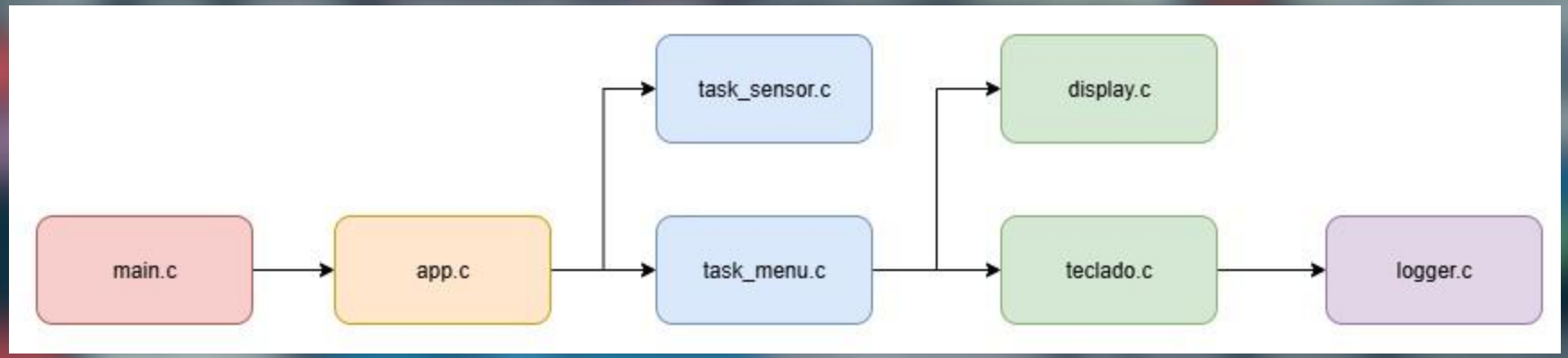


DIAGRAMA DE ESTADO

CODIGOS

- Programacion modular para una mejor organizacion
- Sistema Bare-Metal, sin sistema operativo, del tipo Event-Triggered System
- El codgio main.c inicia el programa y llama a la aplicacion app.c
- Cada codigo cumple una tarea especifica: sensores, menu, display, teclado
- Los modulos se comunican entre si de manera organizada
- El loader carga y prepara el programa para que el sistema pueda ejecutarlo



APP.C

- **Objetivos del codigo:** Gestionar tareas en un sistema bare-metal basado en eventos.
- **Modulos incluidos:** Inicializa y actualiza *task_sensor* y *task_menu*.
- **Estructura de tareas:** Lista *(task_cfg_list)* con punteros a funciones *init* y *update*
- **Variables globales:** Contadores de aplicación y tiempos de ejecución (WCET).
- **Funcion app_init():** Configura variables, inicializa tareas, prepara el contador de ciclos y arranca el sistema.
- **Funcion app_update():** Verifica ticks, ejecuta tareas, mide tiempos y actualiza WCET.
- **Interrupcion HAL_SYSTICK_Callback():** Incrementa contadores para sincronizar ejecución de tareas.

TASK_MENU.C

Objetivos:

- Mostrar fecha, hora y teclas presionadas en el LCD.
- Cambiar entre estados del menú según eventos recibidos de *task_sensor*

Caracteristicas principales:

- Lectura de teclado matricial en tiempo real
- Actualización periódica de fecha y hora simulada.
- Lectura no bloqueante por medio de ticks
- Comunicacion directa con task_sensor via cola de eventos
- Escalable a varios botones sin cambiar la logica

TASK_SENSOR.C

Objetivos:

- Leer botones fisicos y detectar pulsaciones/vibraciones
- Enviar eventos al task_menu utilizando una maquina de estado

Caracteristicas principales:

- Programado en forma Finite State Machine donde el sistema siempre esta en un estado definido
- Ocurre un cambio de estado solo si ocurre un evento valido
- Lectura no bloqueante por medio de ticks
- Debounce por software para evitar rebotes
- Comunicacion directa con task_menu via cola de eventos
- Escalable a varios botones sin cambiar la logica

DISPLAY.C

Objetivos

- Inicializar y controlar un display LCD (4 u 8 bits) para mostrar texto
- Escribir caracteres o cadenas en posiciones específicas del LCD
- Limpiar el display cada vez que se muestra un mensaje nuevo

Caracteristicas principales

- Compatible con conexión GPIO de 4 o 8 bits
- Secuencia de inicialización según protocolo del LCD
- Funcion para posicionar cursor: **displayCharPositionWrite**
- Funcion para escribir texto: **displayStringWrite**
- Control de pines vía **HAL_GPIO_WritePin**
- Temporizacion precisa en microsegundos via **display_delay_us**

TECLADO.C

Objetivos

- Detectar teclas presionadas en un teclado matricial 4x4
- Enviar informacion sobre la tecla presionada a *task_menu*

Caracteristicas principales

- Escaneo por columnas y lectura de filas
- Evita lecturas repetidas hasta soltar tecla
- Funcion para detectar y escanear la letra presionada: **keypadUpdate( )**
- Funcion para consultar la disponibilidad de una tecla: **keypadKeyAvailable( )**
- Funcion para leer la tecla presionada: keypadReadKey**( )**

WCET

- Tiempo maximo de ejecucion de una tarea
- Considera el peor escenario posible
- Se implementa
- Sirve para dimensionar el funcionamiento de temporizadores

IMPLEMENTACION EN FISICO (PMV)

BIBLIOGRAFIA

- **STM32 Arm Programming for Embedded Systems** – Muhammad Ali Mazidi (Author), Shujen Chen (Author), Eshragh Ghaemi (Author)
- **Fundamentals of System-on-Chip Design on Arm Cortex-M Microcontrollers** - René Beuchat (Author), Florian Depraz (Author), Andrea Guerrieri (Author), Sahand Kashani (Author)
- **Programming with STM32: Getting Started with the Nucleo Board and C/C++** – Donald Norris (Author)
- **Embedded C Coding Standard** – Michael Barr (Author)