

# Informe de Proyecto - Cafetera Italiana Automática



Grupo 6 Técnicas Digitales II  
UTN FRBA 2022

Integrantes:

Estudiante	Legajo	Correo
Dimaio, Agustín	160.146-5	adimaio@frba.utn.edu.ar
Guerrero, Tobías	167.273-3	guerrero@frba.utn.edu.ar
Ronchetti, Juan Manuel	173.077-0	jronchetti@frba.utn.edu.ar

## Tabla de contenidos

Tabla de contenidos.....	2
Descripción General.....	3
1.    Introducción.....	3
2.    Funcionamiento básico.....	3
3.    Interfaz con el usuario.....	4
Hardware.....	4
1.    Diagrama en bloques.....	4
2.    Circuito esquemático.....	5
3.    Descripción del circuito.....	5
4.    Circuito impreso.....	6
Software.....	6
1.    Software Microcontrolador.....	6
2.    Sistema operativo.....	8
3.    Librerías.....	8
4.    Programa principal.....	8
5.    Rutinas de interrupción.....	8
6.    Rutinas generales.....	9
Referencias.....	10

## Descripción General

### 1. Introducción

El presente proyecto consta del diseño y construcción de una cafetera tipo italiana (Volturno) automática.

Una cafetera italiana requiere atención por parte del usuario. No sólo se le debe colocar agua y café; también se debe colocar la cafetera sobre una hornalla o anafe, y controlar la temperatura del agua hasta que la misma alcanza el nivel correcto para controlar que el café se haga de forma correcta.

En este proyecto, se propuso automatizar el proceso lo más posible, con un sistema basado en una Blue Pill (STM32F103C6T8) como microcontrolador principal, programado en C.

El dispositivo contará con un anafe, que será el encargado de calentar el agua, y con dos sensores de temperatura, que controlarán la temperatura del agua y la informarán al microcontrolador, que encenderá o apagará el anafe según corresponda en cada momento.

Del lado del usuario se contará con un display que mostrará las opciones seleccionables dentro del menú, entre otra información útil para la utilización de la cafetera. El menú será navegable mediante tres botones, cada uno con una función específica. Además, la cafetera contará con indicaciones para el usuario mediante LEDs y un buzzer.

### 2. Funcionamiento básico

La cafetera cuenta con dos modos de funcionamiento: café rápido y programar café. El modo de café rápido inicia un proceso de realización de café en el momento en que se lo selecciona. En cambio, el modo de programar café le ofrece al usuario la posibilidad de configurar una hora a la cual se realice el café automáticamente. Para esto, se requiere que la cafetera esté encendida hasta la hora de realizar el proceso, además de que debe tener todos los ingredientes preparados.

En ambos casos, el proceso sucede de la misma forma. El microcontrolador prende el anafe y comienza a censar la temperatura mediante los sensores. Cuando la temperatura leída (correspondiente a la temperatura del agua) alcanza el nivel necesario para que se haga correctamente el café, se apaga el anafe. En este momento, inicia un temporizador cuyo valor fue determinado mediante pruebas, al final del cual se considera finalizado el proceso. Se cuenta con un buzzer, leds y el display para dar aviso al usuario.

En todo momento, el microcontrolador está capacitado para determinar algún error en los sensores de temperatura, automáticamente apagando el anafe para evitar accidentes.

### 3. Interfaz con el usuario

La cafetera cuenta con un display LCD de dos líneas. El menú funciona mediante su propia máquina de estados, donde según el estado de la máquina principal, en el display se muestra el mensaje correspondiente.

El menú es navegable por el usuario, mediante tres botones: uno para aceptar (OK), uno para cancelar (CANCEL), y uno de opciones (SEL).

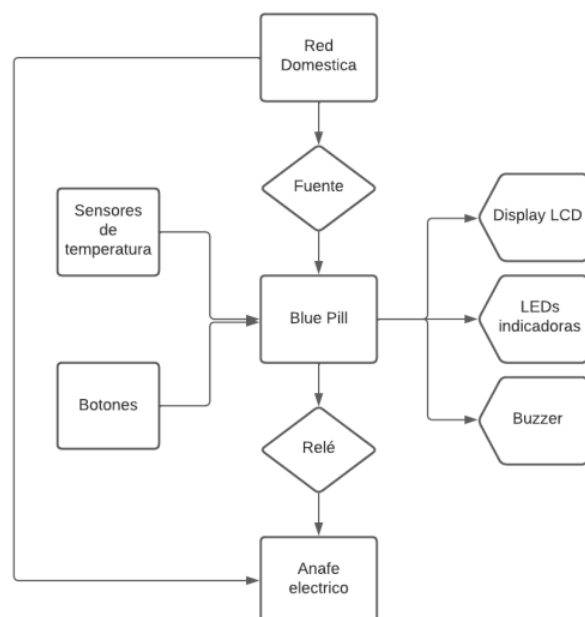
Al iniciar la cafetera, en el display se muestra un mensaje indicando que la misma está iniciando. Una vez iniciado el sistema, aparece en el display el menú principal. En este, se muestran dos opciones correspondientes a los dos modos de funcionamiento de la cafetera: “Café Rápido” y “Programar Café”. Cada opción aparece en una línea distinta, con espacio para un cursor que indica cuál es la opción seleccionada. El botón SEL permite cambiar la opción seleccionada, y el botón OK selecciona la opción.

Al seleccionar la opción “Programar Café”, se avanza a un menú en el cual se permite configurar la hora. El botón SEL permite modificar hora (cada 1 hora) y minutos (cada 10 minutos), el botón OK confirma la programación del proceso para la hora seleccionada, y el botón CANCEL permite volver al menú anterior. Si se confirma el horario, al llegar al mismo comenzará el proceso, pasando el display a mostrar el tiempo estimado restante para que el café esté listo.

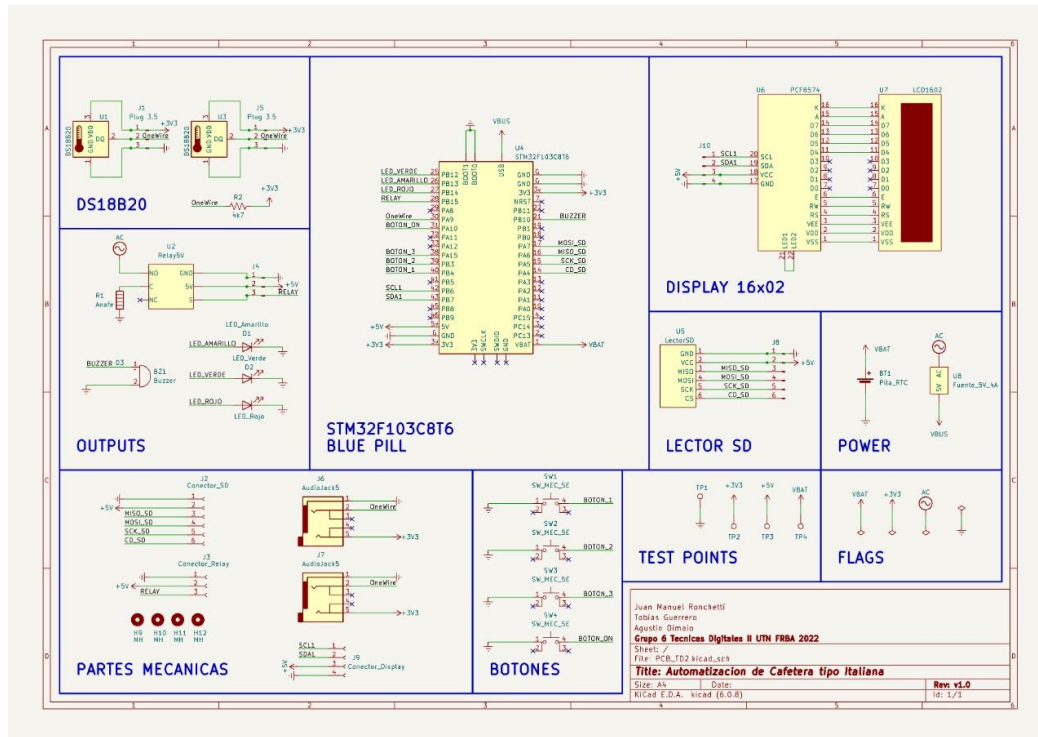
En cambio, si se selecciona la opción “Café Rápido”, se pasa directamente a mostrar el tiempo estimado restante para que el café esté listo al igual que en el caso anterior.

## Hardware

### 1. Diagrama en bloques



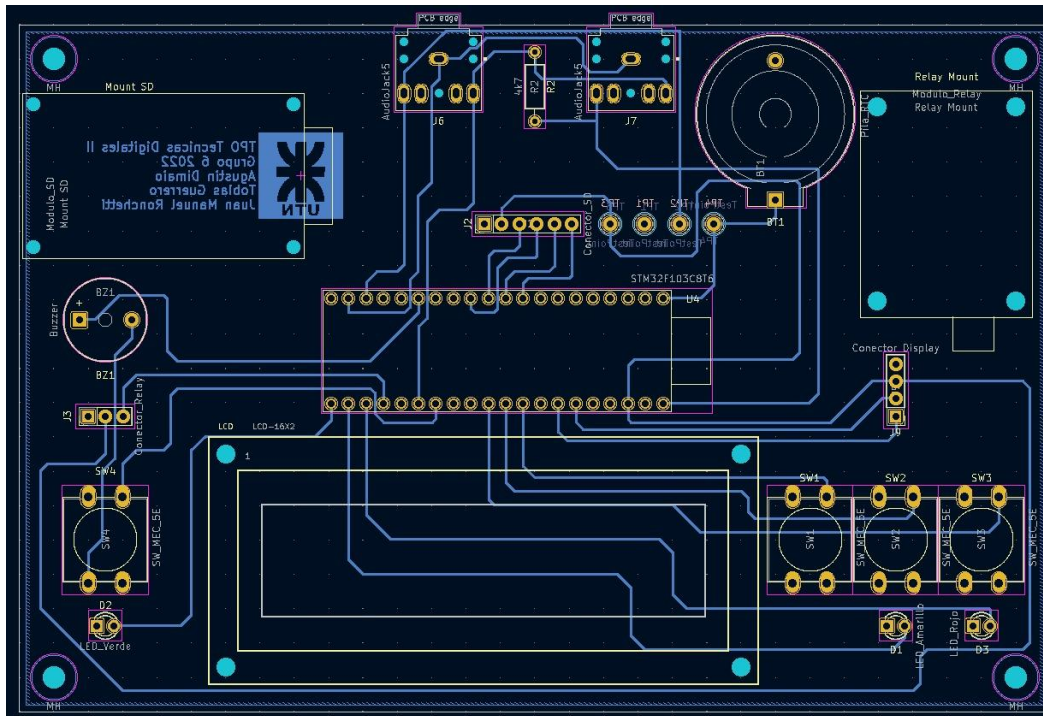
## 2. Circuito esquemático



## 3. Descripción del circuito

- **DS18B20:** sensores de temperatura utilizados para el control del agua. Se colocan dos unidades para poder asegurarse el correcto funcionamiento de los mismos y que no se provoquen sobrecalentamiento del agua. Los dos pines DQ (One Wire) de cada sensor se conectan a una resistencia de pull-up con la finalidad de mantenerlo siempre en un modo alto o HIGH.
- **OUTPUTS:** se muestra el buzzer como se utiliza como indicador sonoro, los LED's que se utilizan como indicadores visuales y el módulo de relay junto con el anafe eléctrico para el control del mismo, ya que se alimenta de la red eléctrica de 220v.
- **PARTES MECANICAS:** pines designados para la conexión del módulo SD, del módulo de relay, los jacks de 3.5mm utilizados como conectores rápidos para los sensores y por último el conector del display.
- **STM32F103C8T6 BLUE PILL:** microcontrolador utilizado para el proyecto junto con sus conexiones.
- **BOTONES:** botones que servirán para el control de la cafetera.
- **DISPLAY 16x02:** display LCD utilizado para dar indicaciones de los estados de la cafetera.
- **LECTOR SD:** módulo de lector de SD en donde se guardará información relacionada al funcionamiento de los sensores y el anafe.
- **POWER:** fuente switching de 5V 4A utilizada para la alimentación del proyecto junto con la batería para el uso del RTC.
- **TEST POINTS:** puntos de prueba diseñados para el PCB.
- **FLAGS:** señalizaciones utilizadas en el esquemático.

## 4. Circuito impreso



## Software

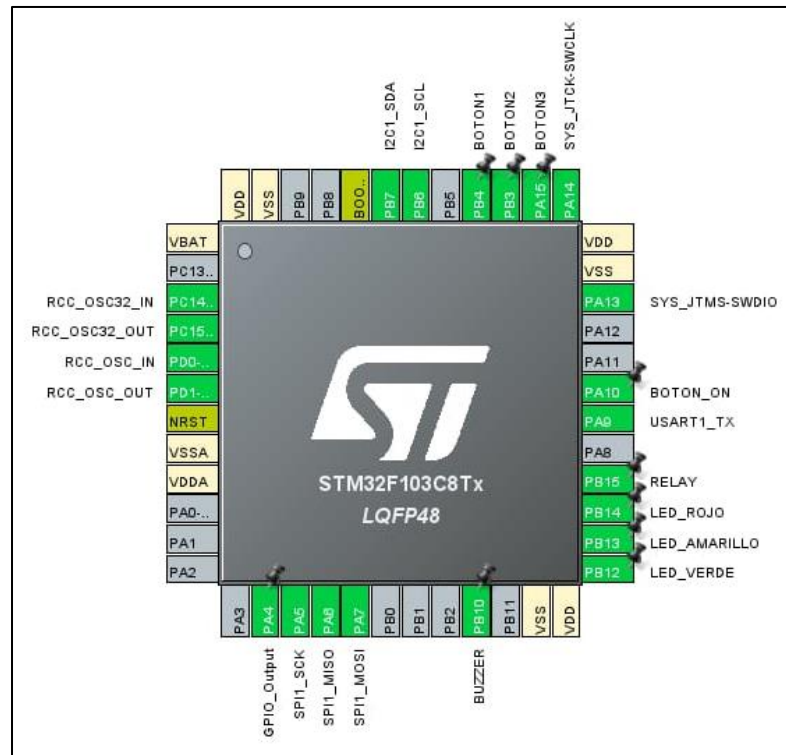
### 1. Software Microcontrolador

A continuación, se mencionan los recursos utilizados:

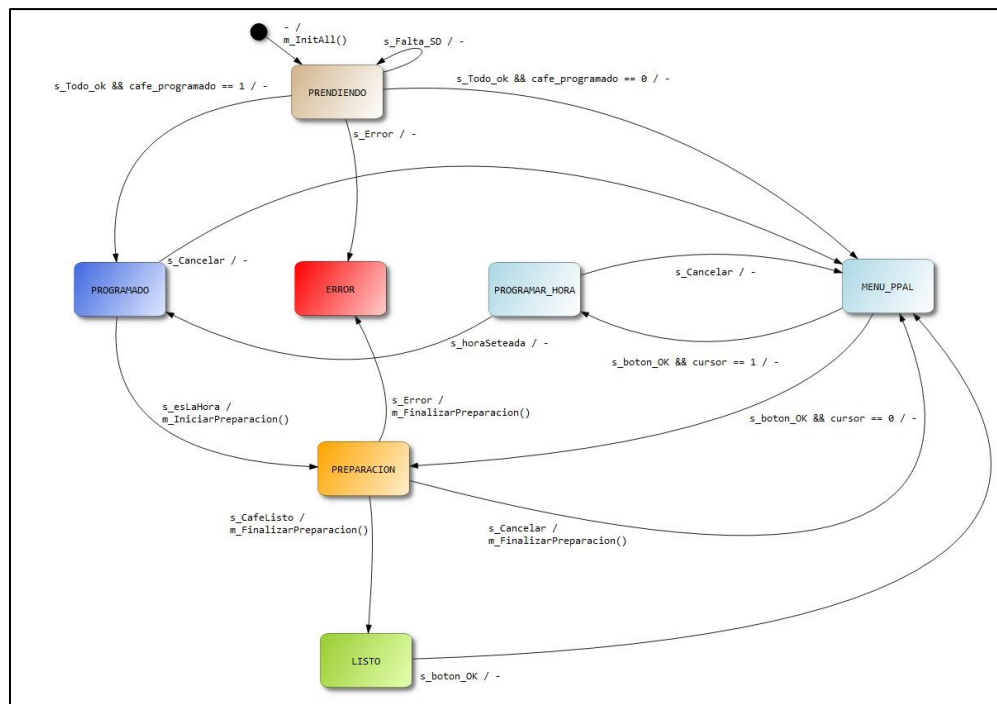
- RTC interno sin output del STM32.
- Timer TIM4 para base de tiempo del sistema (scheduler).
- Clocks HSE y LSE con Crystal Resonator.
- Pines B6 y B7 para con I2C1 para la conexión del display.
- Pin A9 para la USART1 en configuración Half-Duplex Single-Wire para los sensores de temperatura.
- Pines B4, B3, A10 y A15 como input con pull-up interno para los botones 1, 2, 3 y 4.
- Pines B10, B12, B13, B14, B15 como outputs push-pull para respectivamente el buzzer, LED Verde, LED Azul, LED Rojo y el Relay.
- Pines A4, A5, A6 y A7 con SP1 en modo Full-Duplex master (middleware FATFS) para la SD.

Entorno de desarrollo: STM32CubeIDE.

IOC:



Máquina de estados principal:



## 2. Sistema operativo

El sistema operativo utilizado es FreeRTOS.

## 3. Librerías

A continuación, se listan las librerías utilizadas:

- LCD Liquid Cristal I2C (Nizar Mohideen)
- One wire STM32 (Stanislav Lakhtin)
- File Handling - basada en fatfs y file\_handling\_RTOS (Controllers Tech)

Al final del documento en el apartado de referencias se encuentran los links a cada una.

## 4. Programa principal

- Comenzamos realizando la configuración del clock y las inicializaciones necesarias para los GPIO, I2C, USART1, RTC y la del Kernel del FreeRTOS.

```
SystemClock_Config();  
MX_GPIO_Init();  
MX_I2C1_Init();  
MX_USART1_UART_Init();  
MX_RTC_Init();  
InitAll();  
osKernelInitialize();
```

- Luego procedemos a la creación de las tareas de FreeRTOS:

```
xTaskCreate(MdE_Principal, "MdE"...  
xTaskCreate(DriverTeclado, "Driver Teclado"...  
xTaskCreate(Tarea_Lectura, "Lectura Temperatura"...  
xTaskCreate(Error_Sensores, "Error Sensores"...  
xTaskCreate(Error_Anafe, "Error Anafe"...  
xTaskCreate(Tarea_Timer, "Tarea Timer"...  
xTaskCreate(Tarea_Timer2, "Tarea Timer 2"...
```

- Por último, iniciamos el Kernel de FreeRTOS para dar lugar a la planificación y ejecución de las tareas mediante la lógica de la MdE:

```
osKernelStart();
```

## 5. Rutinas de interrupción

Solo se utilizan las rutinas de interrupción propias del Kernel de FreeRTOS.



## 6. Rutinas generales

A continuación, se listan las tareas de FreeRTOS utilizadas:

- MdE principal: encargada del funcionamiento general del sistema.
- Driver teclado: encargada de la detección de los botones y el antirrebote.
- Lectura temperatura: lee la temperatura de los sensores.
- Timer y Timer 2: timers de software implementados gracias al RTC.
- Error anafe: tarea de detección de error en el anafe.
- Error sensores: tarea de detección de error en los sensores de temperatura.

## Referencias

- FreeRTOS: <https://www.freertos.org/>
- STM32CubeIDE: <https://www.st.com/en/development-tools/stm32cubeide.html>
- Software de la cafetera: [https://github.com/micuentz1/TPO\\_TD2](https://github.com/micuentz1/TPO_TD2)
- LCD Liquid Cristal I2C (Nizar Mohideen): [https://www.youtube.com/watch?v=JvnlJPao&ab\\_channel=MicroPetabyNizarMohideen](https://www.youtube.com/watch?v=JvnlJPao&ab_channel=MicroPetabyNizarMohideen)
- One wire STM32 (Stanislav Lakhtin): <https://github.com/StanislavLakhtin/stm32-DS18B20>
- File Handling - basada en fatfs y file\_handling\_rtos (Controllers Tech): [https://www.youtube.com/watch?v=spVIZO-jbxE&ab\\_channel=ControllersTech](https://www.youtube.com/watch?v=spVIZO-jbxE&ab_channel=ControllersTech)