

# CORE Interface

- [CORE API Documentation - Core8-16F](#)

# CORE API Documentation - Core8-16F

The CORE object is an instance of the CORE16F\_System\_Interface\_t structure, which provides a collection of core functionalities available in the CORE8-16F framework. Below is a brief overview of the key functions and features available through CORE.

## CORE Definition

```
const CORE16F_System_Interface_t CORE = {  
    .Initialize = &CORE16F_init,  
  
    #ifdef _CORE16F_SYSTEM_INCLUDE_DELAYS_ENABLE  
        .Delay_MS = &CORE16F_Delay_BlockingMS,  
    #endif / _CORE16F_SYSTEM_INCLUDE_DELAYS_ENABLE/  
  
    #ifdef _CORE16F_SYSTEM_EVENTS_ENABLE  
        .Events_Initialize = &TimedEventSystem_Init,  
        .Events_Add = &ScheduleEvent,  
        .Events_Check = &CheckEvents,  
        .Events_Remove = &CancelEvent,  
    #endif  
  
    .Make16 = &CORE_Make_16,  
    .Low4 = &CORE_Return_4bit_Low,  
    .High4 = &CORE_Return_4bit_High,  
    .Set_Bit = &CORE_Set_Bit,  
    .Clear_Bit = &CORE_Clear_Bit,  
    .FloatToString = &CORE_floatToString,  
    .IntToString = &CORE_intToString,  
};
```

## Key Functionalities of CORE

- **Initialization:** `CORE.Initialize()` initializes the core system, including clock and peripheral setup.
- **Delays** (Conditional): If enabled, `CORE.Delay_MS(timeMS)` provides a blocking delay in milliseconds.
- **Event Management** (Conditional): If enabled, CORE offers several event management functions:
  - **CORE.Events\_Initialize():** Initializes the event management system, allowing events to be scheduled and handled.
  - **CORE.Events\_Add(delay\_ms, callback, interval):** Schedules an event to occur after a specified delay in milliseconds. The callback function will be called when the event occurs, and interval specifies if the event should repeat (0 for a one-time event).
  - **CORE.Events\_Check():** Checks for any pending events and executes their associated callbacks if the conditions are met.
  - **CORE.Events\_Remove(callback):** Cancels a previously scheduled event by providing the callback function associated with it. This is useful for stopping recurring events or removing unwanted scheduled events.
- **Utility Functions:** Various helper functions are provided:
  - `CORE.Make16(high_byte, low_byte):` Combines two 8-bit values into a 16-bit value.
  - `CORE.Low4(byte)`, `CORE.High4(byte):` Extracts lower or higher 4 bits from an 8-bit value.
  - `CORE.Set_Bit(byte, bit_position)`, `CORE.Clear_Bit(byte, bit_position):` Sets or clears a specific bit in an 8-bit value.
  - `CORE.FloatToString(number, buffer, decimalPlaces):` Converts a float to a string representation.
  - `CORE.IntToString(number, buffer):` Converts an integer to a string representation.

The CORE object provides a convenient and centralized interface for accessing key functions of the CORE8-16F system, making it easier for developers to initialize, control, and manage the microcontroller's features effectively.