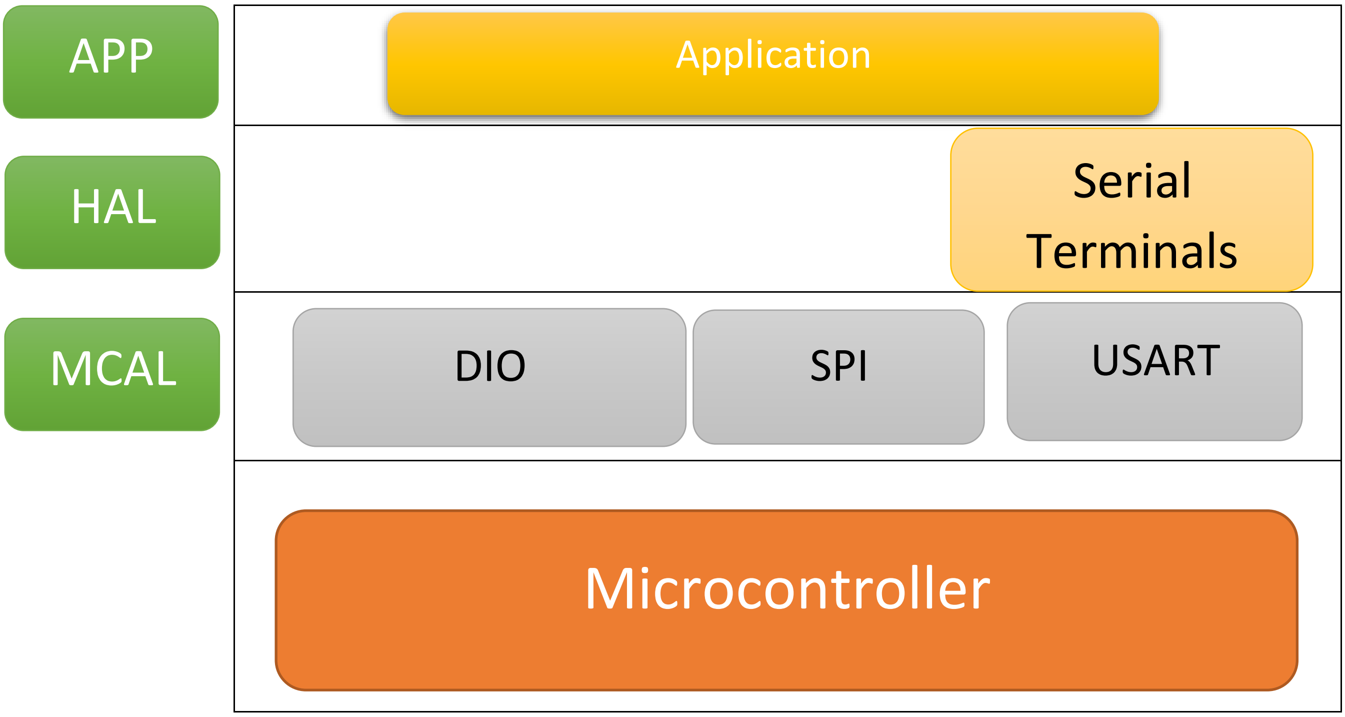


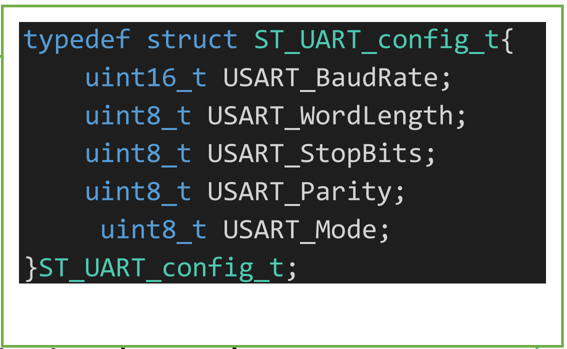
1. MCAL

1.1. DIO APIs

```
/* This Function determine the Pin direction Whether input or output */
E_status  DIO_SetPinDirection(ST_DIO_config_t * Configurations);
typedef ST_DIO_config_t{
                          EN_portname_t port_name;
                          EN_portno_t pin_no;
                          EN_pinstate_t state;                // input or output
}ST_DIO_config_t;
E_status  DIO_SetPinValue(uint8_t au8_port_no, uint8_t au8_pin_no, uint8_t au8_value;);
E_status  DIO_GetPinValue(uint8_t au8_port_no, uint8_t au8_pin_no, uint8_t * data);
E_status  DIO_TogglePin(uint8_t au8_port_no, uint8_t au8_pin_no);
```

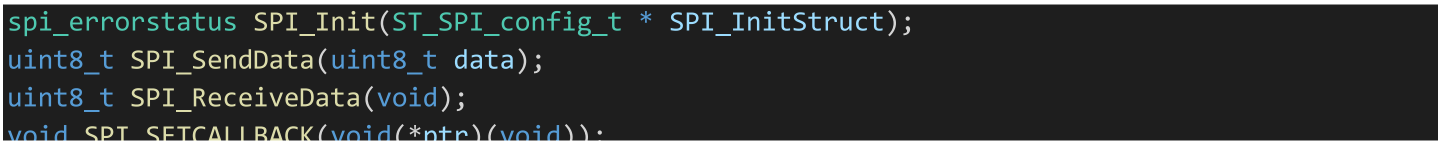
1.2. Usart APIs

```
error_state usart_Init(ST_UART_config_t * USART_InitStruct);

error_state usart_SendData(uint8_t data_transmitted);

uint8_t usart_ReceiveData(void);

error_state usart_SendString(uint8_t *str);

uint8_t * usart_ReceiveString(uint8_t * au8data ,uint8_t terminating_character);
```

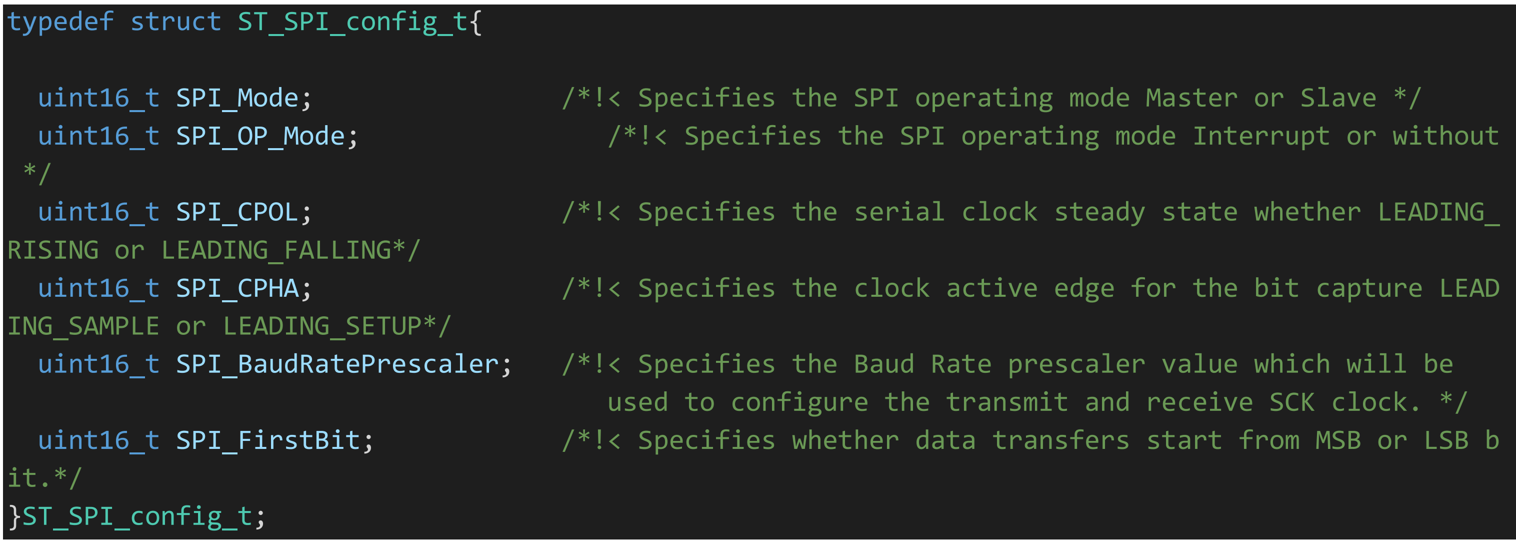


```
typedef struct ST_UART_config_t{
    uint16_t USART_BaudRate;
    uint8_t USART_WordLength;
    uint8_t USART_StopBits;
    uint8_t USART_Parity;
     uint8_t USART_Mode;
}ST_UART_config_t;
```

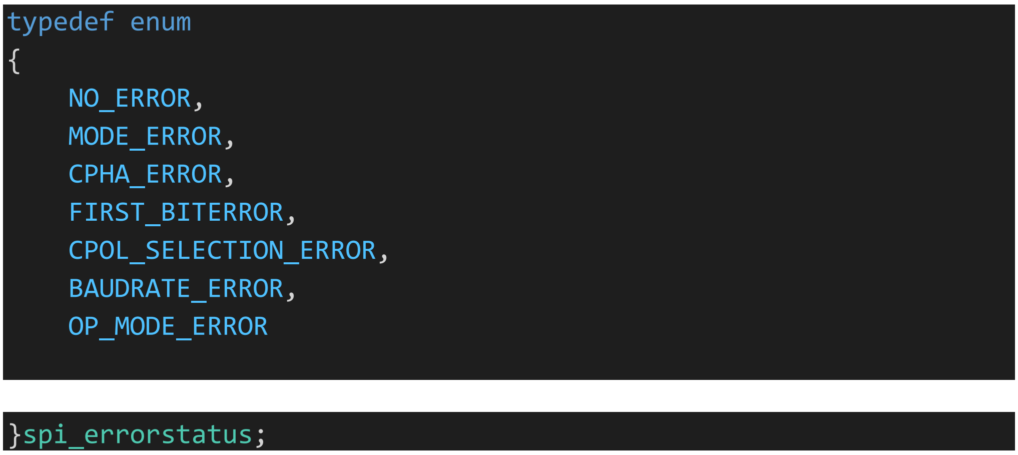
1.3. SPI APIs



```c
spi_errorstatus SPI_Init(ST_SPI_config_t * SPI_InitStruct);
uint8_t SPI_SendData(uint8_t data);
uint8_t SPI_ReceiveData(void);
void SPI_SETCALLBACK(void(*ptr)(void));
```



```c
typedef struct ST_SPI_config_t{

  uint16_t SPI_Mode;                  /*!< Specifies the SPI operating mode Master or Slave */
  uint16_t SPI_OP_Mode;                  /*!< Specifies the SPI operating mode Interrupt or without
 */
  uint16_t SPI_CPOL;                  /*!< Specifies the serial clock steady state whether LEADING_
RISING or LEADING_FALLING*/
  uint16_t SPI_CPHA;                  /*!< Specifies the clock active edge for the bit capture LEAD
ING_SAMPLE or LEADING_SETUP*/
  uint16_t SPI_BaudRatePrescaler;     /*!< Specifies the Baud Rate prescaler value which will be
                                         used to configure the transmit and receive SCK clock. */
  uint16_t SPI_FirstBit;              /*!< Specifies whether data transfers start from MSB or LSB b
it.*/
}ST_SPI_config_t;
```



```c
typedef enum
{
    NO_ERROR,
    MODE_ERROR,
    CPHA_ERROR,
    FIRST_BITERROR,
    CPOL_SELECTION_ERROR,
    BAUDRATE_ERROR,
    OP_MODE_ERROR

}spi_errorstatus;
```

2. HAL

a. EEPROM

```
void EEPROM_VoidInit(void);
void EEPROM_VoidWriteDataByte(uint8_t SlaveAddress , uint8_t InternalReg, uint8_t Data);
uint8_t EEPROM_u8ReadDataByte(uint8_t SlaveAddress , uint8_t InternalReg);
```