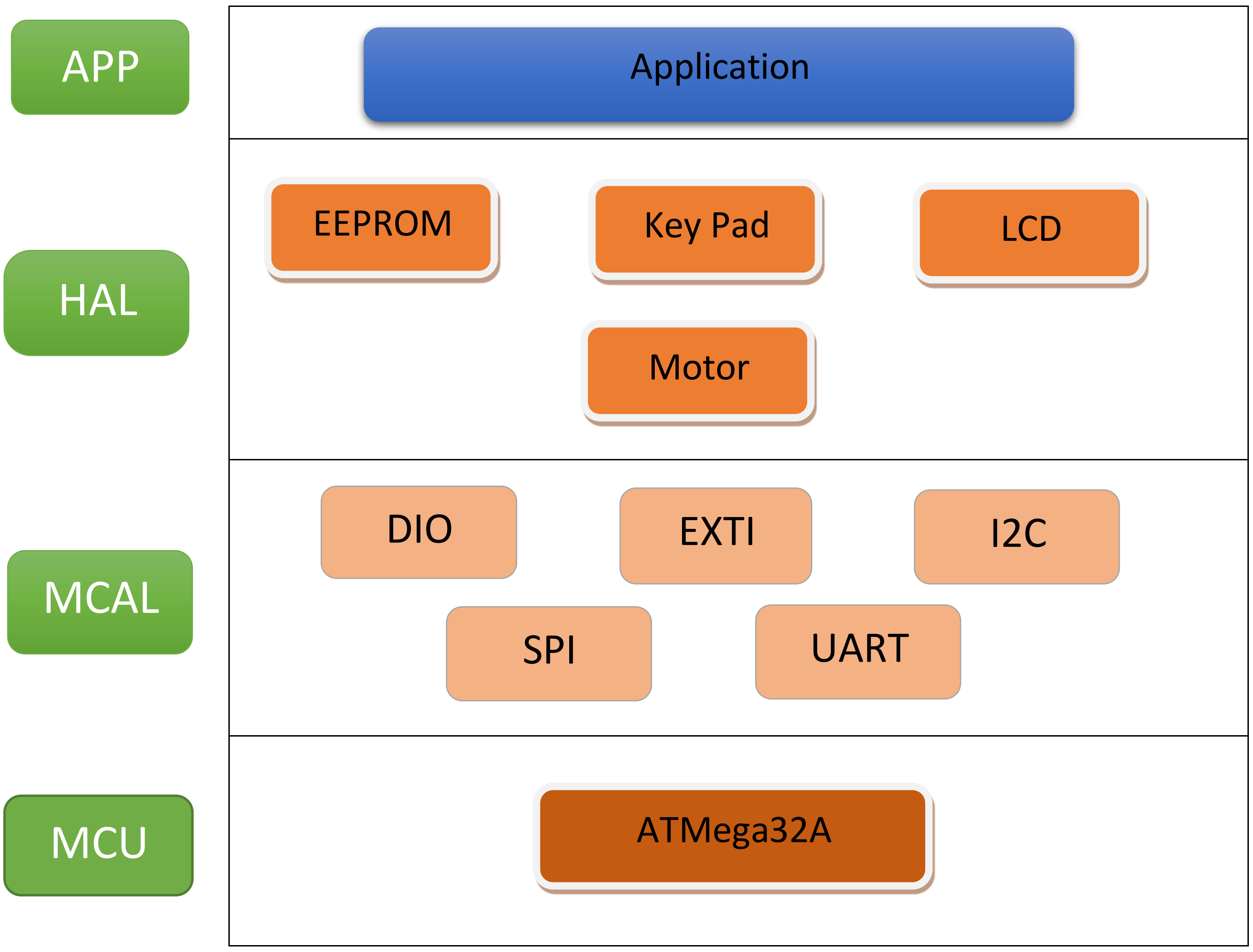
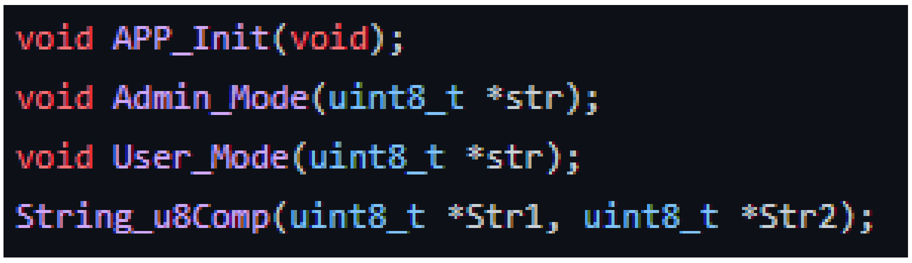


1.Application

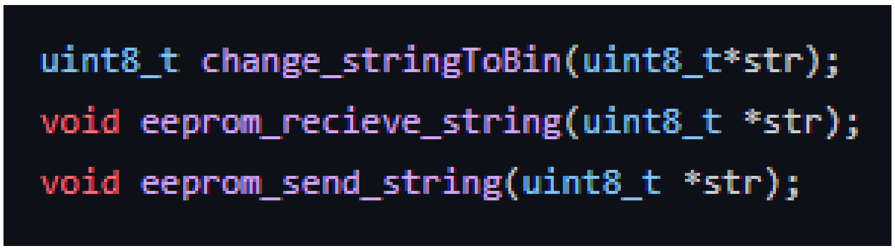
1.1.APP APIs



```
void APP_Init(void);
void Admin_Mode(uint8_t *str);
void User_Mode(uint8_t *str);
String_u8Comp(uint8_t *Str1, uint8_t *Str2);
```

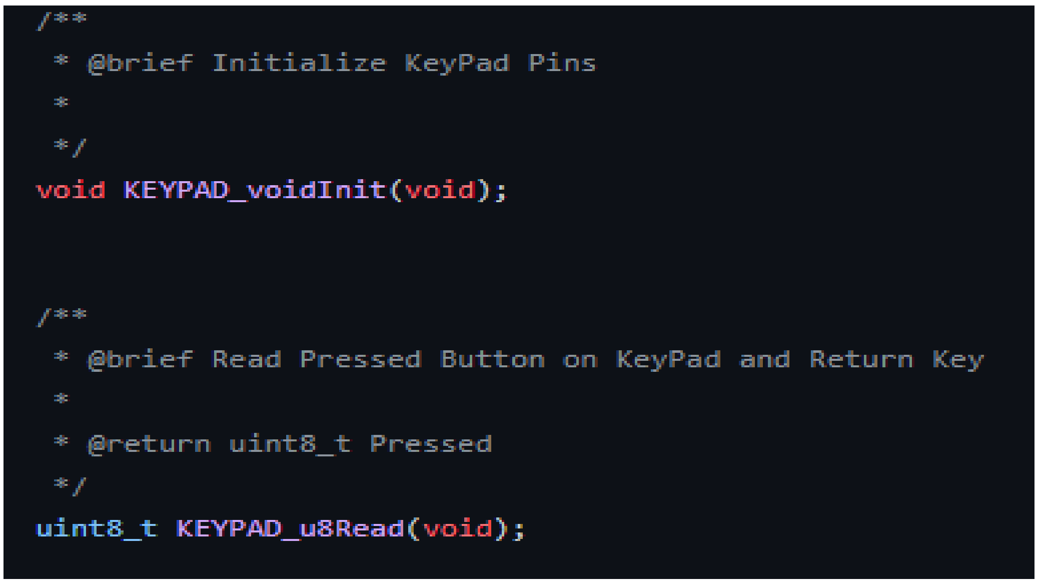
2.HAL

2.1.EEPROM APIs



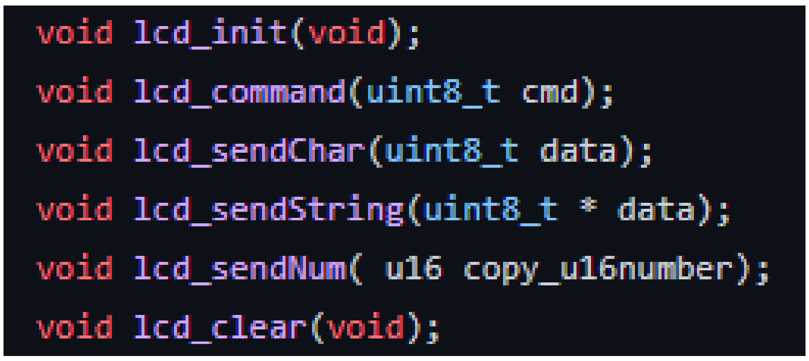
```c
uint8_t change_stringToBin(uint8_t*str);
void eeprom_recieve_string(uint8_t *str);
void eeprom_send_string(uint8_t *str);
```

2.2.Key Pad



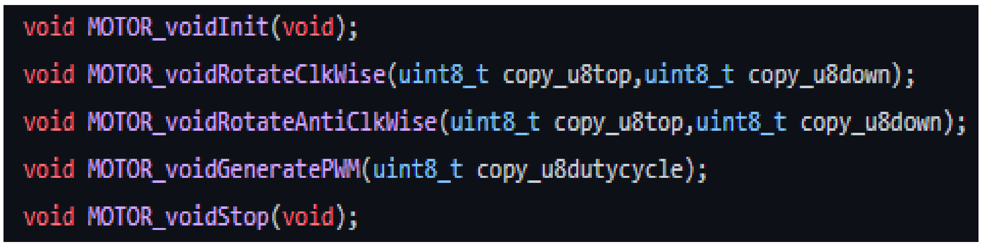
```c
/**
 * @brief Initialize KeyPad Pins
 *
 */
void KEYPAD_voidInit(void);


/**
 * @brief Read Pressed Button on KeyPad and Return Key
 *
 * @return uint8_t Pressed
 */
uint8_t KEYPAD_u8Read(void);
```

2.3.LCD



```c
void lcd_init(void);
void lcd_command(uint8_t cmd);
void lcd_sendChar(uint8_t data);
void lcd_sendString(uint8_t * data);
void lcd_sendNum( u16 copy_u16number);
void lcd_clear(void);
```

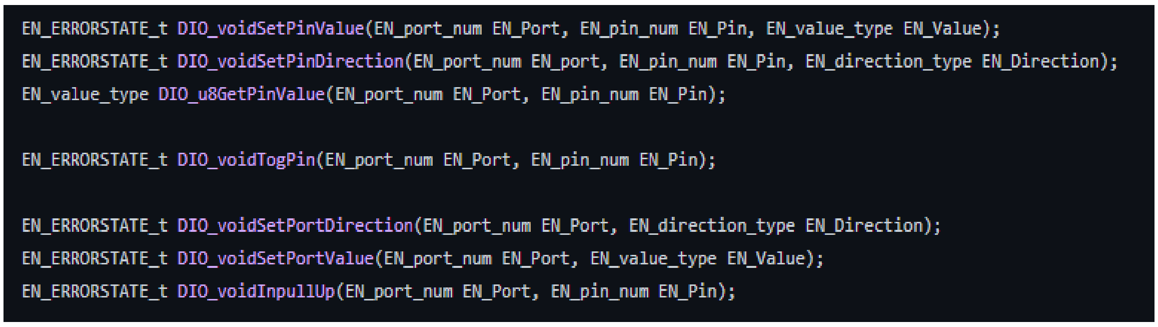
2.4.Motor



```c
void MOTOR_voidInit(void);
void MOTOR_voidRotateClkWise(uint8_t copy_u8top,uint8_t copy_u8down);
void MOTOR_voidRotateAntiClkWise(uint8_t copy_u8top,uint8_t copy_u8down);
void MOTOR_voidGeneratePWM(uint8_t copy_u8dutycycle);
void MOTOR_voidStop(void);
```

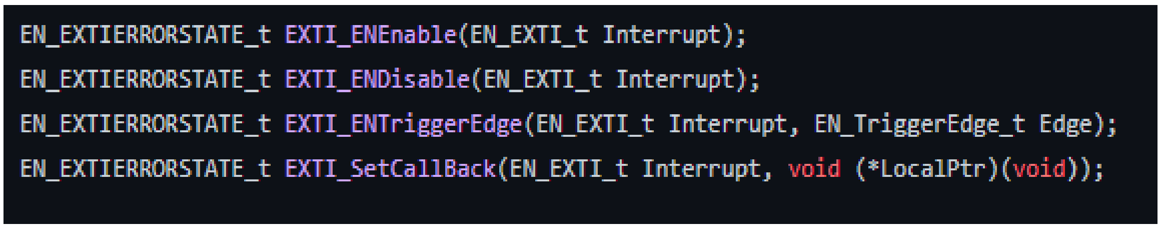
3.MCAL

3.1.DIO



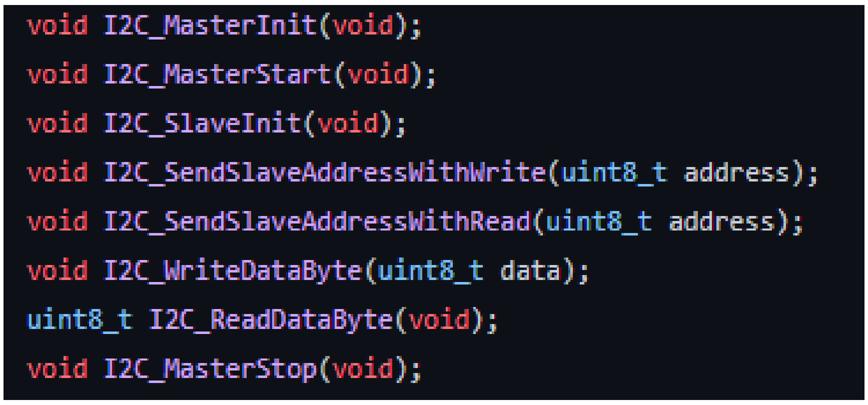
```c
EN_ERRORSTATE_t DIO_voidSetPinValue(EN_port_num EN_Port, EN_pin_num EN_Pin, EN_value_type EN_Value);
EN_ERRORSTATE_t DIO_voidSetPinDirection(EN_port_num EN_port, EN_pin_num EN_Pin, EN_direction_type EN_Direction);
EN_value_type DIO_u8GetPinValue(EN_port_num EN_Port, EN_pin_num EN_Pin);

EN_ERRORSTATE_t DIO_voidTogPin(EN_port_num EN_Port, EN_pin_num EN_Pin);

EN_ERRORSTATE_t DIO_voidSetPortDirection(EN_port_num EN_Port, EN_direction_type EN_Direction);
EN_ERRORSTATE_t DIO_voidSetPortValue(EN_port_num EN_Port, EN_value_type EN_Value);
EN_ERRORSTATE_t DIO_voidInpullUp(EN_port_num EN_Port, EN_pin_num EN_Pin);
```

3.2.EXTI



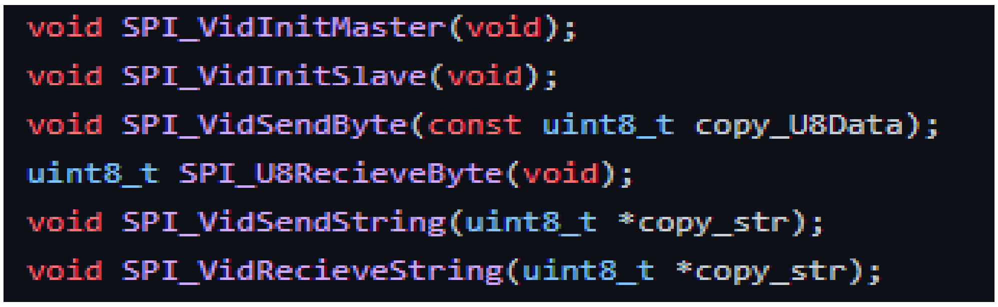
```c
EN_EXTIERRORSTATE_t EXTI_ENEnable(EN_EXTI_t Interrupt);
EN_EXTIERRORSTATE_t EXTI_ENDisable(EN_EXTI_t Interrupt);
EN_EXTIERRORSTATE_t EXTI_ENTriggerEdge(EN_EXTI_t Interrupt, EN_TriggerEdge_t Edge);
EN_EXTIERRORSTATE_t EXTI_SetCallBack(EN_EXTI_t Interrupt, void (*LocalPtr)(void));
```

3.3.I2C



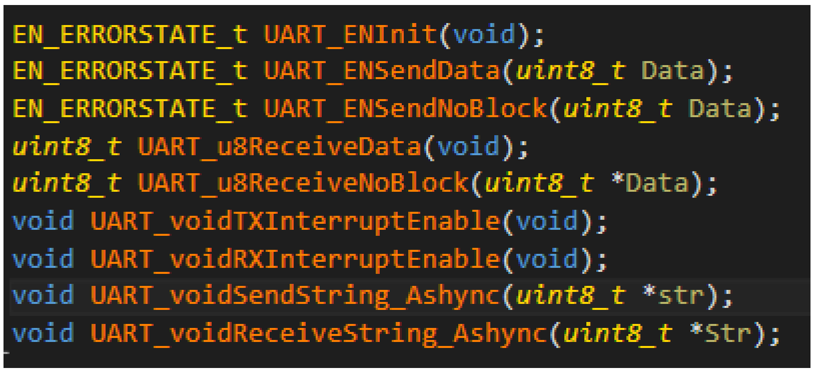
```c
void I2C_MasterInit(void);
void I2C_MasterStart(void);
void I2C_SlaveInit(void);
void I2C_SendSlaveAddressWithWrite(uint8_t address);
void I2C_SendSlaveAddressWithRead(uint8_t address);
void I2C_WriteDataByte(uint8_t data);
uint8_t I2C_ReadDataByte(void);
void I2C_MasterStop(void);
```

3.4.SPI



```
void SPI_VidInitMaster(void);
void SPI_VidInitSlave(void);
void SPI_VidSendByte(const uint8_t copy_U8Data);
uint8_t SPI_U8RecieveByte(void);
void SPI_VidSendString(uint8_t *copy_str);
void SPI_VidRecieveString(uint8_t *copy_str);
```

3.5.UART



```
EN_ERRORSTATE_t UART_ENInit(void);
EN_ERRORSTATE_t UART_ENSendData(uint8_t Data);
EN_ERRORSTATE_t UART_ENSendNoBlock(uint8_t Data);
uint8_t UART_u8ReceiveData(void);
uint8_t UART_u8ReceiveNoBlock(uint8_t *Data);
void UART_voidTXInterruptEnable(void);
void UART_voidRXInterruptEnable(void);
void UART_voidSendString_Ashync(uint8_t *str);
void UART_voidReceiveString_Ashync(uint8_t *Str);
```