Air Conditioning System

Team 2

1- Mohab Ahmed

2- Anas Mahmoud

3- Mustafa Mohammed

4- Omar Taha

Table of Contents

1-Description

1.1 Hardware components

1. LCD (2*16)

2. Keypad (3 * 3) (Note: 4 buttons will be used)

3. Temperature sensor (LM35)

4. Buzzer

1.2 software Requirements

1. The system starts initially by welcoming the user

2. Then inform the user about the default temp and ask for desired one

3. Display range of temperature starts from min 18 and to max 35 C

4. To enter the desired temperature user has access through two buttons button one to increase and button two to decrease above or below the default temperature

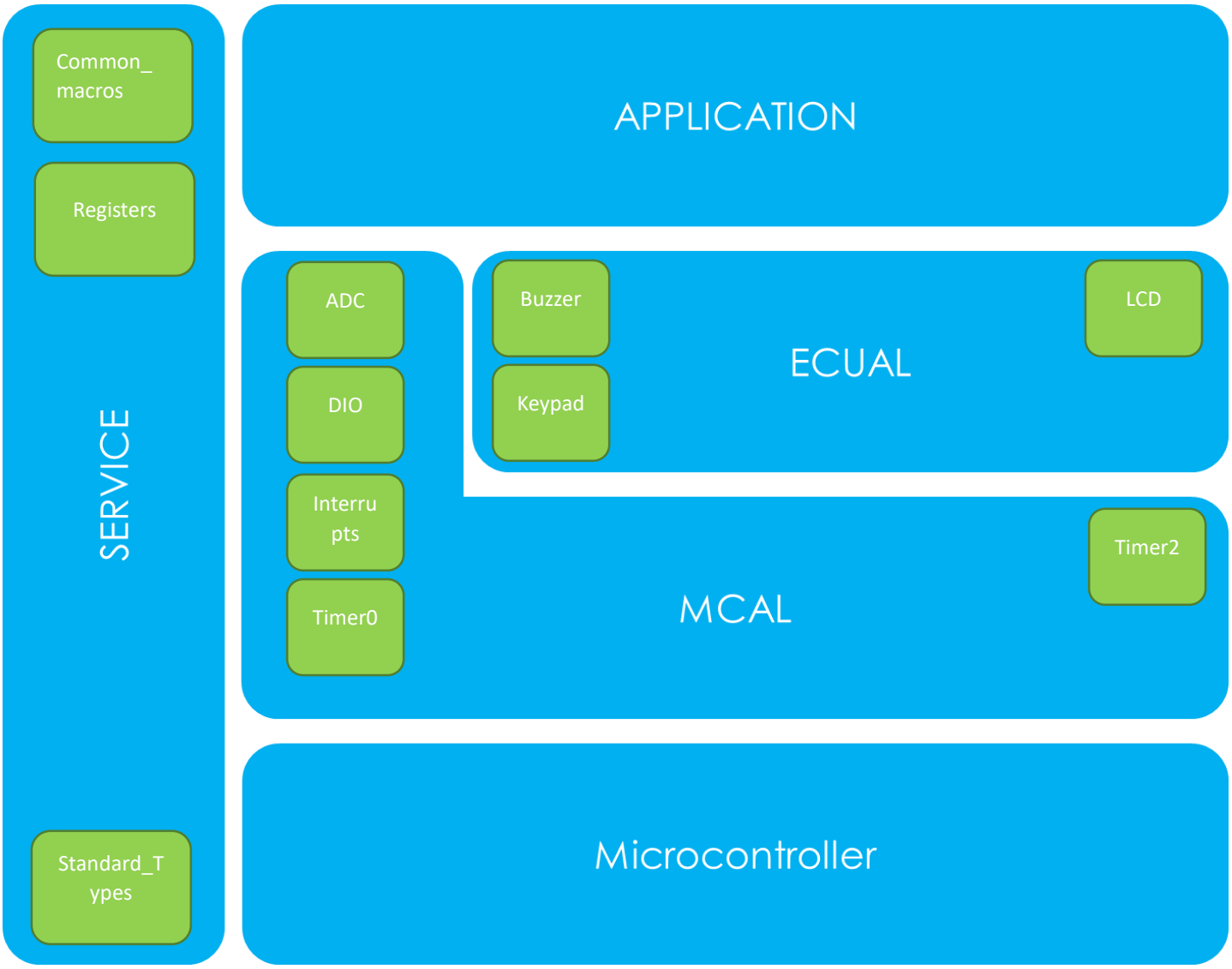
5. Once button three is pressed the desired temperature will be set

6. If the desired temperature is greater than the default one buzzer will be on until both are equal

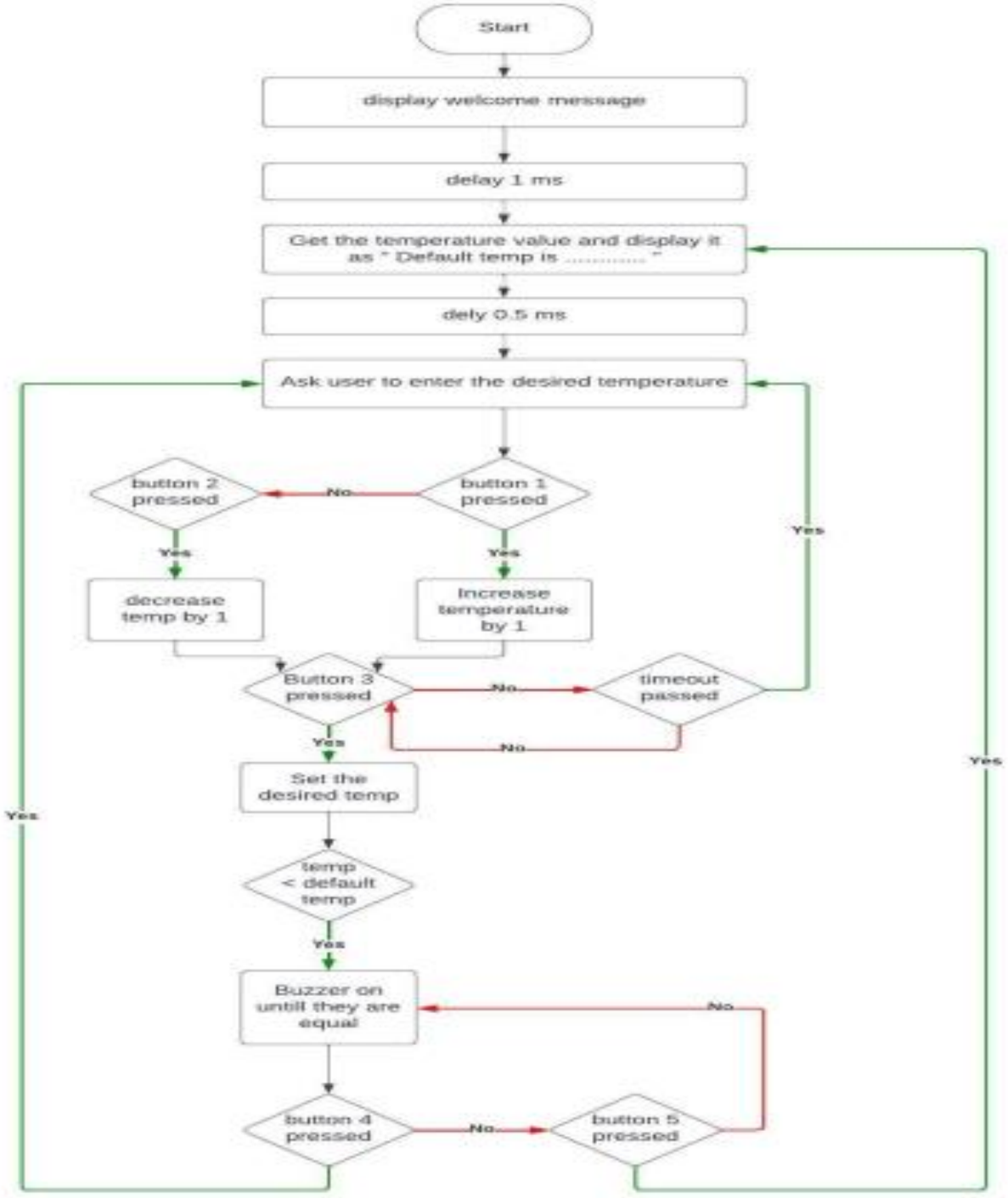
7. If button 4 is pressed at any time that will take the user back to set another desired temperature

8. Button 5 will be used to reset the system to start again

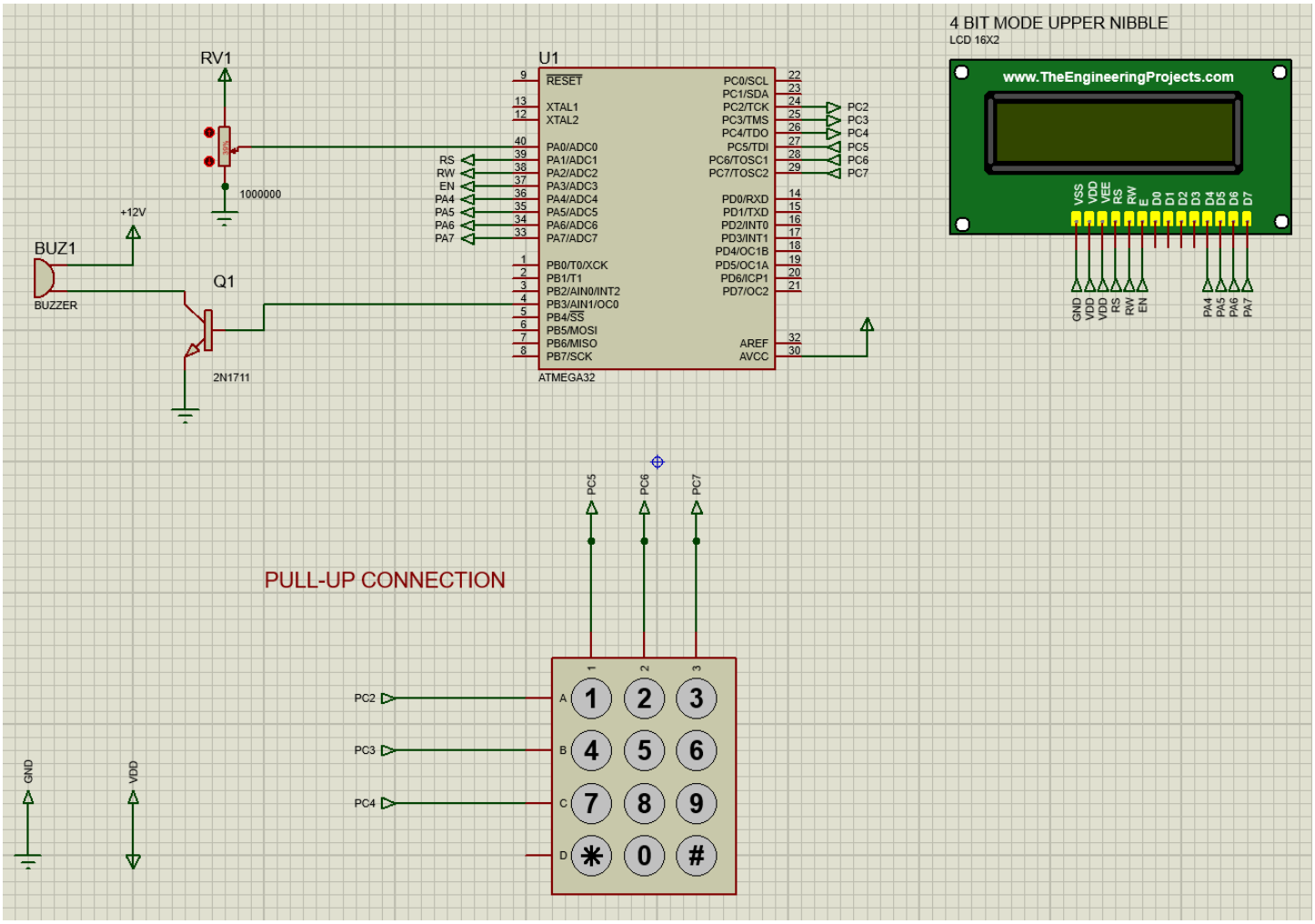
2-Layered architecture



3-System Flow Chart



4-Schematic Capture



5-Drivers Description

5.1 DIO Driver

Configuration: Consist of 4 API's

Location: MCAL

Function: used to set pin direction (input or output), pin value (high or low) or read a value from a pin or toggle a pin

5.2 Timer Driver

Configuration: Consist of 5 API's

Location: MCAL

Function: used to set a time delay

5.3 ADC Driver

Configuration: Consist of 2 API's

Location: MCAL

Function: used to initialize ADC, read the value of ADC

5.4 Keypad Driver

Configuration: Consist of 2 API's

Location: HAL

Function: used to initialize the keypad, get pressed key

5.5 LCD Driver

Configuration: Consist of 14 API's

Location: HAL

Function: used to initialize the LCD, send command to LCD & display character or string to LCD & jump to specific position on LCD & to clear the LCD & to wright integer or float number on the LCD

5.6 Application Driver

Configuration: Consist of 9 API's

Location: App

Function: combine between the drivers API's to meet the requirement

6-API's

6.1 DIO Driver

1- PinDirection_t DIO_setpindir (uint8_t u8_a_portid, uint8_t u8_a_pinid,

uint8_t u8_a_pindir);

2- PinValue_t DIO_setpinvalue (uint8_t u8_a_portid, uint8_t u8_a_pinid, uint8_t u8_a_pinval);

3- PinRead_t DIO_readpin (uint8_t u8_a_portid, uint8_t u8_a_pinid, uint8_t* u8_a_val);

4- PinRead_t DIO_togglepin (uint8_t u8_a_portid, uint8_t u8_a_pinid);

6.2 Timer0 Driver

1- TMR0_init_error TMR0_init(void);

2- TMR0_start_error TMR0_start(void);

3- TMR0_stop_error TMR0_stop(void);

4- TMR0_delay_error TMR0_delayms(uint32_t u32_a_delayms);

5- TMR0_delay_error TMR0_delaymicos(uint32_t u32_a_delaymicros);

6.3 Timer2 Driver

1- err_state TIMER2_normalMode(void);

2- err_state TIMER2_initialValue(uint8_t value);

3- err_state TIMER2_perscalerMode(unsigned int prescaler);

4- err_state TIMER2_delay(float f_a_delayInMillis);

5- unsigned int TIMER2_getInitialValue(float f_a_delayInMillis);

6.4 ADC Driver

1- ADC_initstatus ADC_Init(void);

2- uint16_t ADC_read(void);

6.5 Keypad Driver

1- void KEYPAD_init(void) ;

2- uint8_t KEYPAD_getpressedkey(void) ;

6.6 LCD Driver

1- LCD_init_error LCD_8_bit_init (void);

2- LCD_sendCommand_error LCD_8_bit_sendCommand(uint8_t u8_a_command);

3- LCD_sendChar_error LCD_8_bit_sendChar(uint8_t u8_a_char);

4- LCD_init_error LCD_4_bit_init(void);

5- LCD_sendCommand_error LCD_4_bit_sendCommand(uint8_t u8_a_command);

6- LCD_sendChar_error LCD_4_bit_sendChar(uint8_t u8_a_char);

7- LCD_sendString_error LCD_sendString(uint8_t *u8_a_string);

8- void LCD_goTo(uint8_t u8_a_row,uint8_t u8_a_column);

9- void LCD_createCustomCharacter(uint8_t *u8_a_bitMap,uint8_t u8_a_location)

10- LCD_init_error LCD_init(void);

11- LCD_sendCommand_error LCD_sendCommand(uint8_t u8_a_command);

12- LCD_sendChar_error LCD_sendChar(uint8_t u8_a_char);

13- LCD_sendChar_error LCD_sendFloat(float f_a_number);

14- LCD_sendChar_error LCD_sendInteger(uint16_t u16_a_number);

6.7 APP Driver

1- APP_initError APP_init(void);

2-void APP_printString(uint8_t u8_a_row, uint8_t u8_a_column, uint8_t *u8_a_str);

3-void APP_printChar(uint8_t u8_a_row, uint8_t u8_a_column, uint8_t u8_a_char);

4-void APP_printInteger(uint8_t u8_a_row, uint8_t u8_a_column, uint16_t u16_a_integer);

5-void APP_setTemp();

6-void APP_decrementBar();

7-void APP_incrementBar();

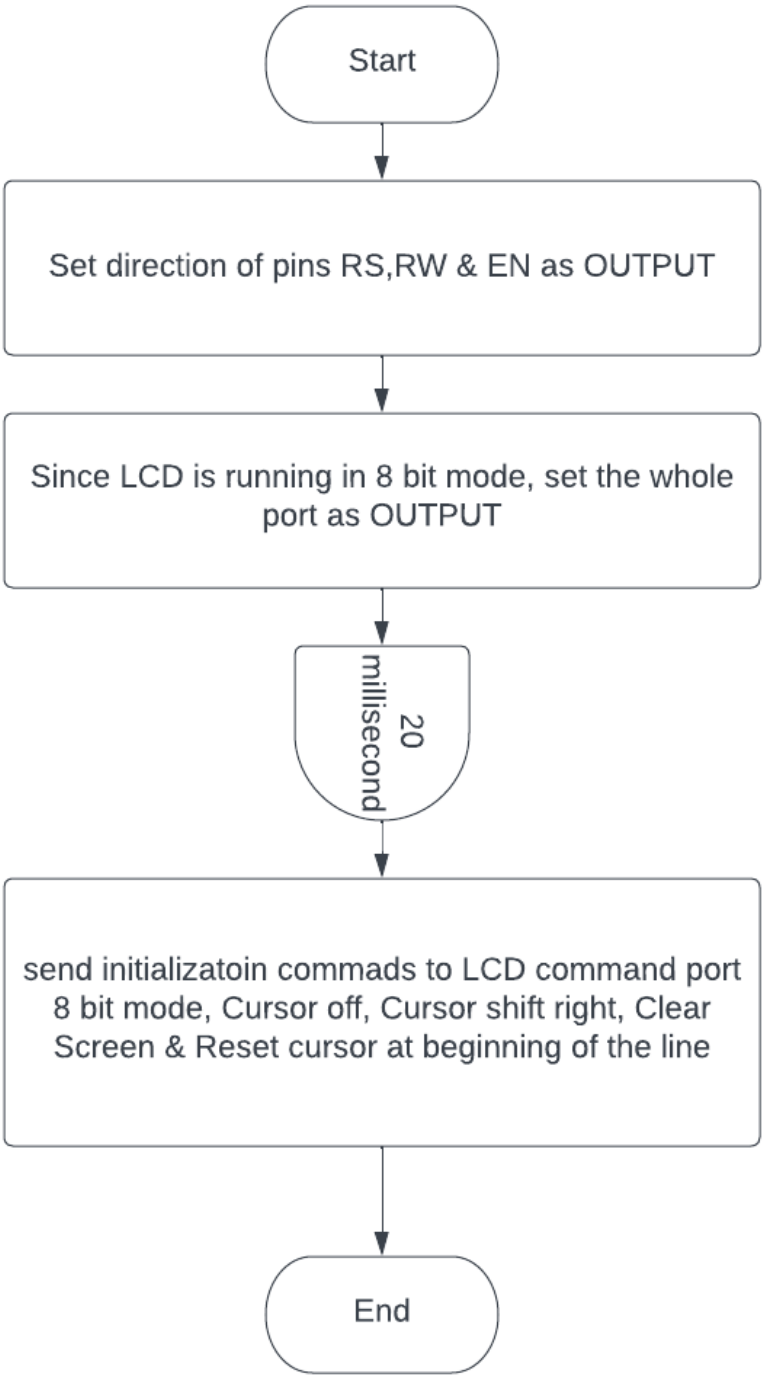
8-void APP_defaultView(void);

9- void get_current_overflow(void);

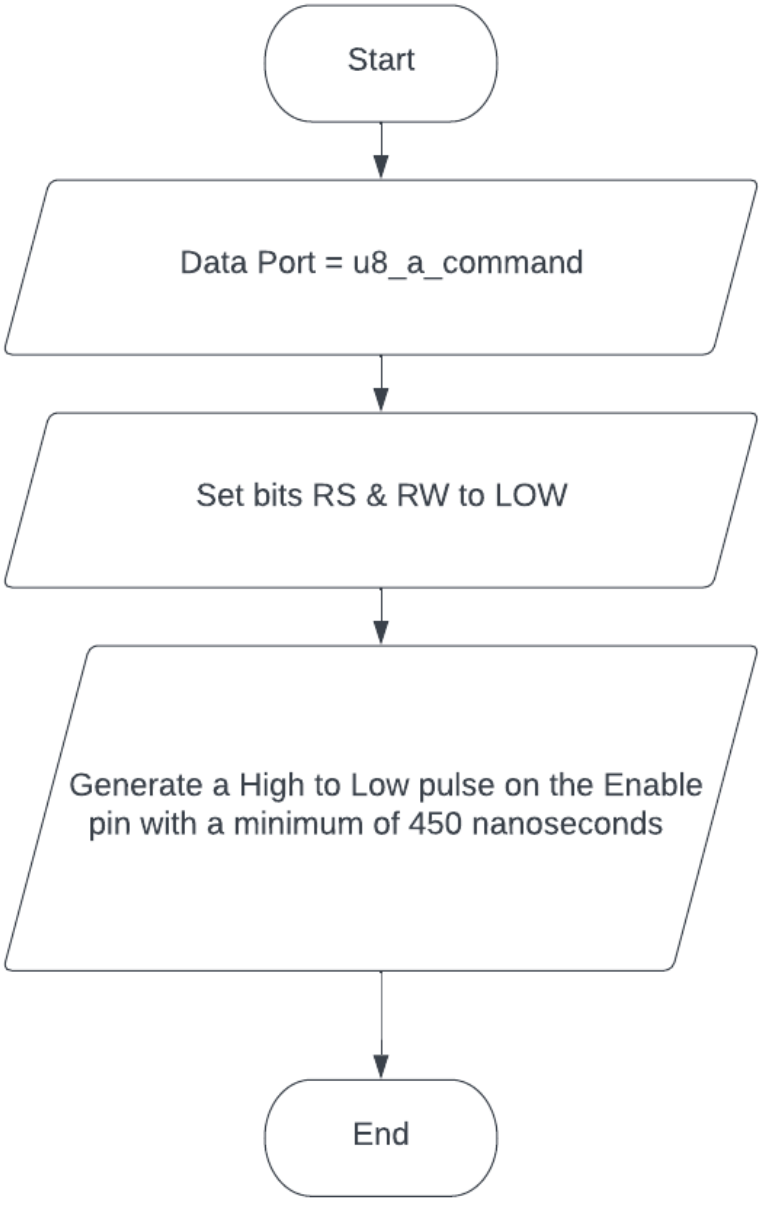
7- API's Flow Chart

7.1 LCD

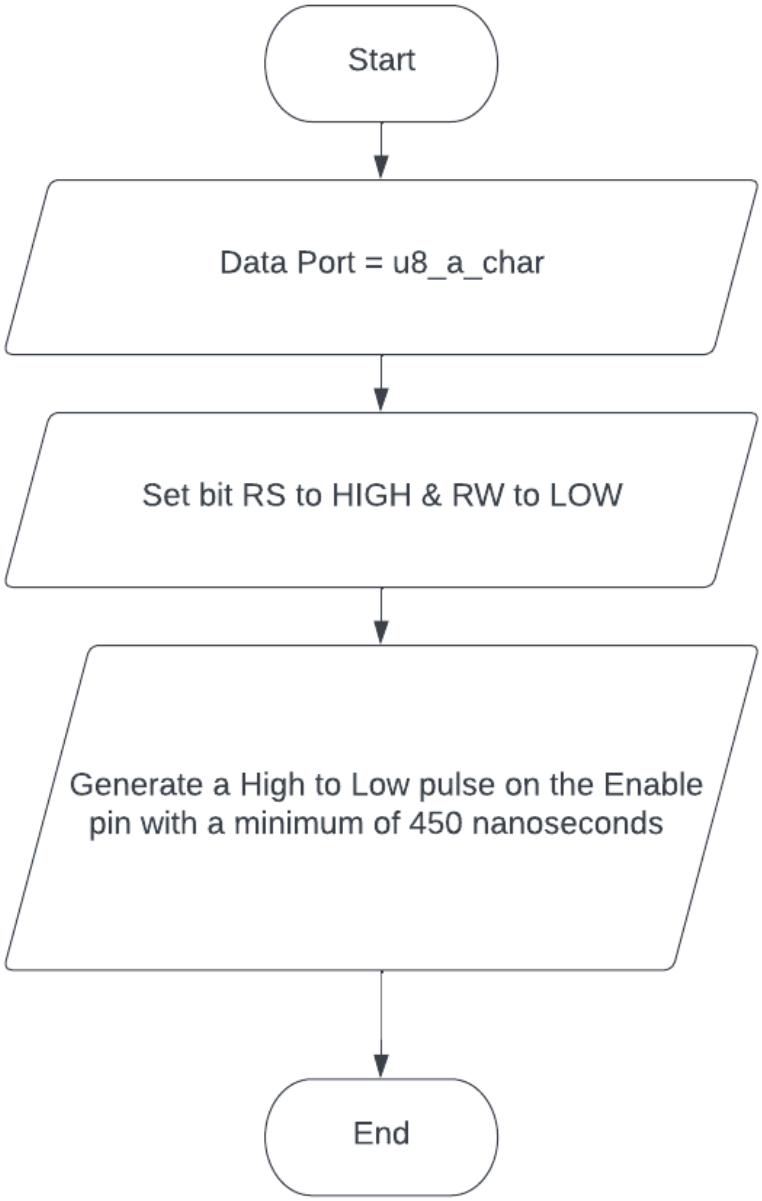
```
LCD_init_error LCD_8_bit_init(void);
```



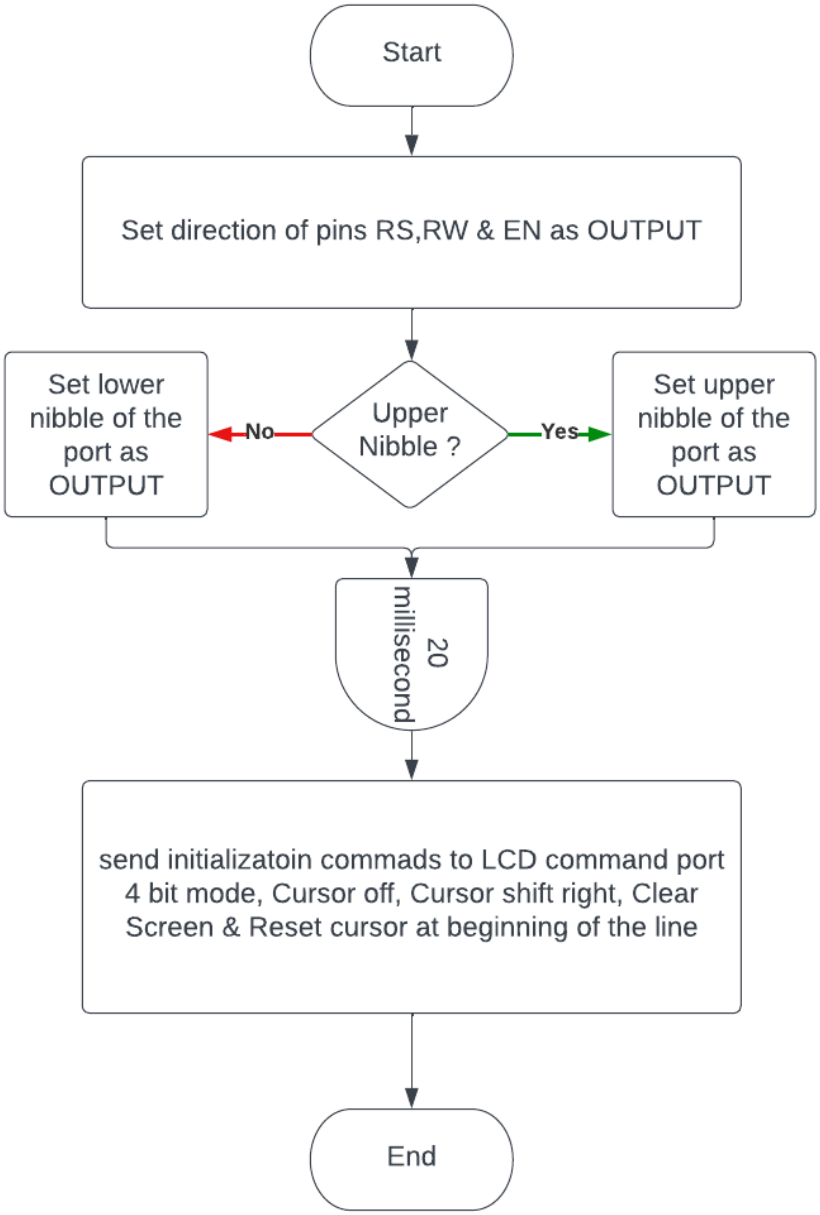
```
LCD_sendCommand_error LCD_8_bit_sendCommand(uint8_t u8_a_command);
```



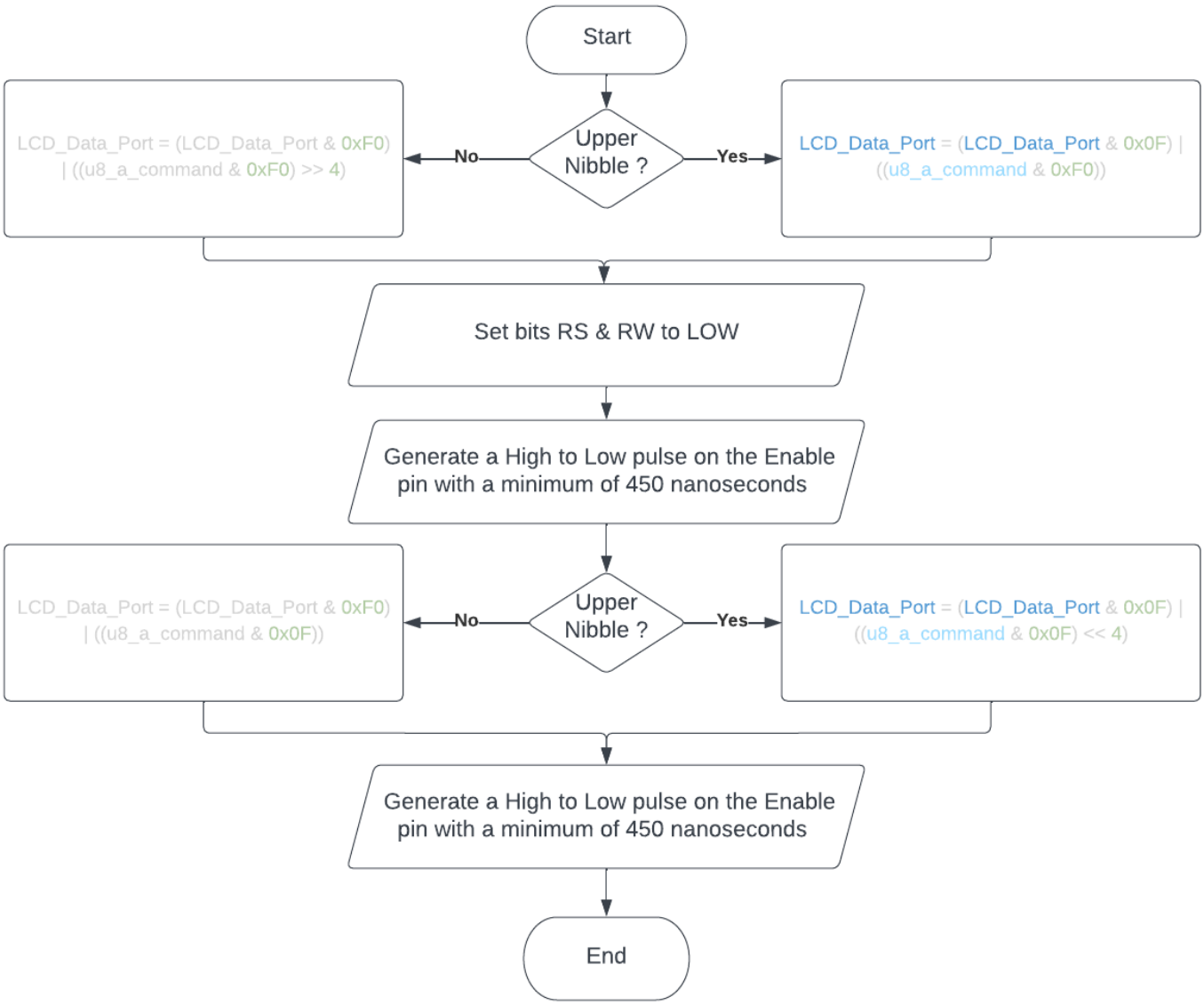
```
LCD_sendChar_error LCD_8_bit_sendChar(uint8_t u8_a_char);
```



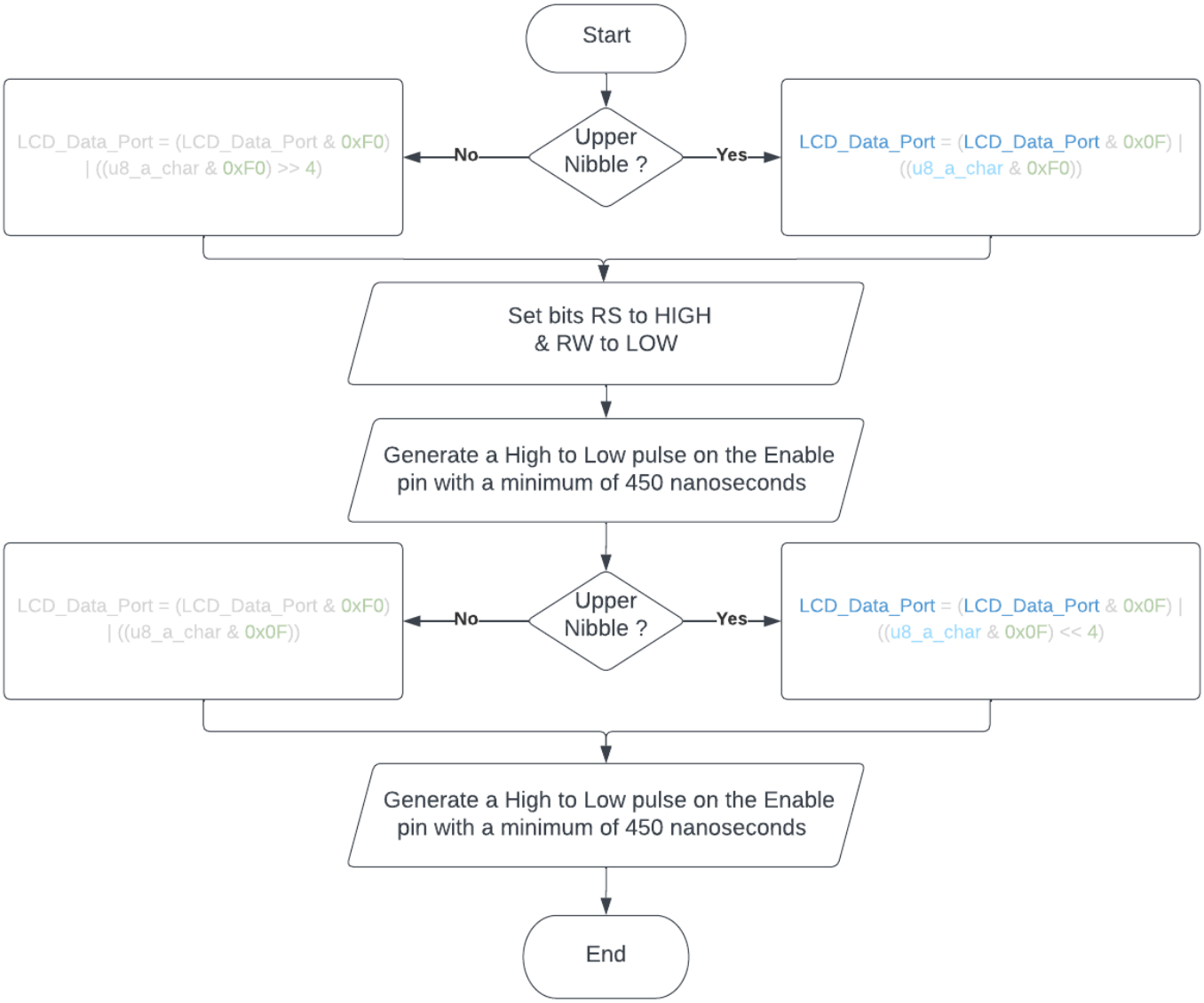
```
LCD_init_error LCD_4_bit_init(void);
```



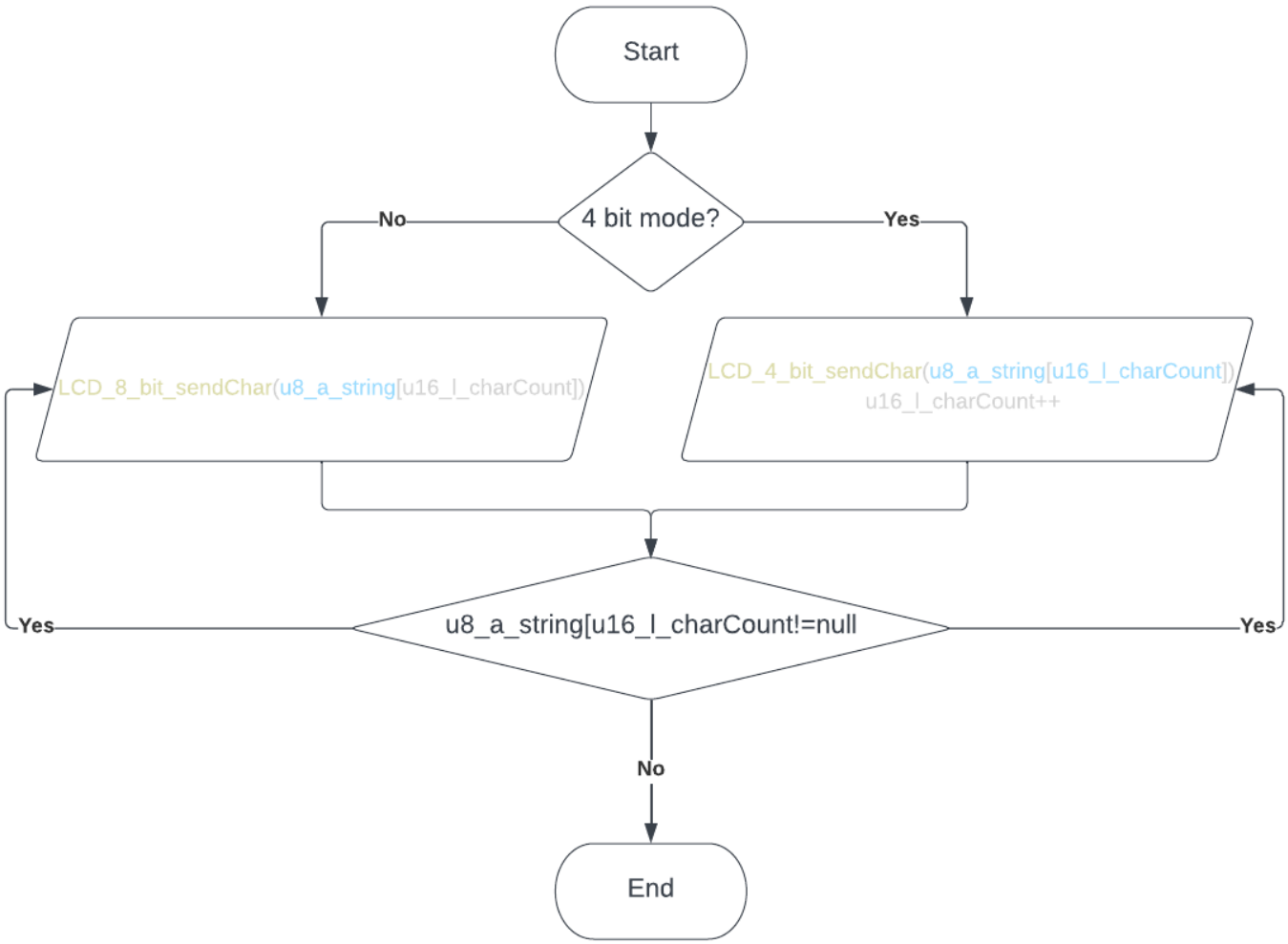
```
LCD_sendCommand_error LCD_4_bit_sendCommand(uint8_t u8_a_command);
```



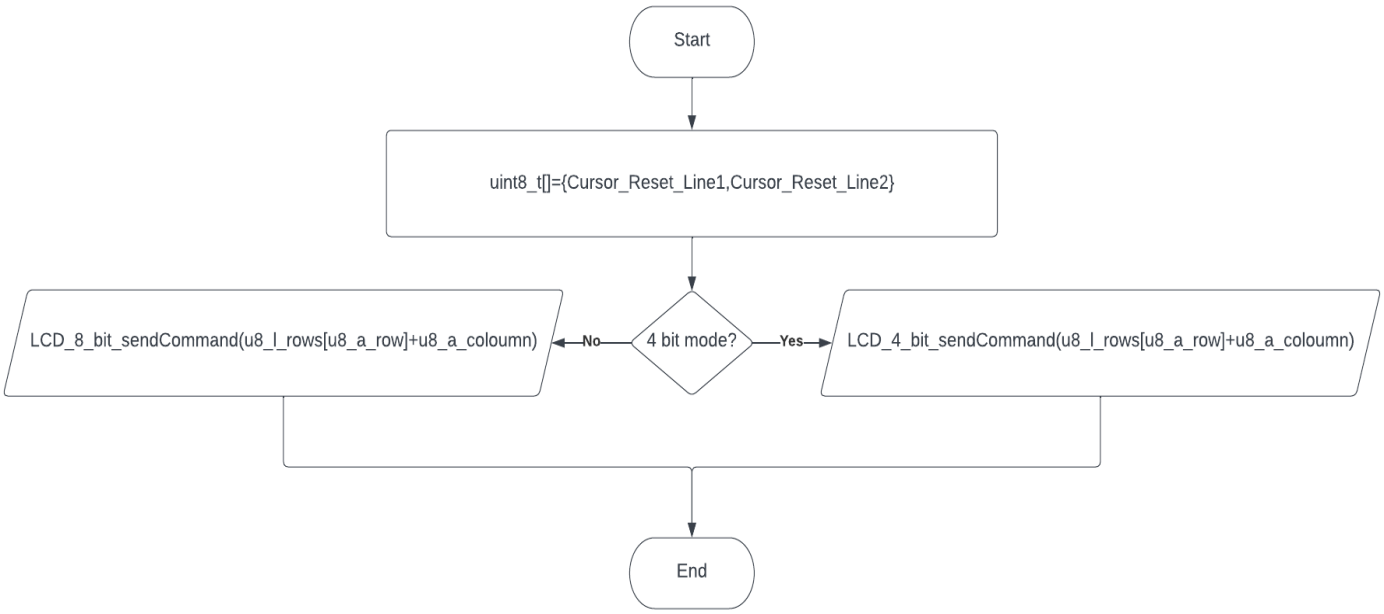
```
LCD_sendChar_error LCD_4_bit_sendChar(uint8_t u8_a_char);
```



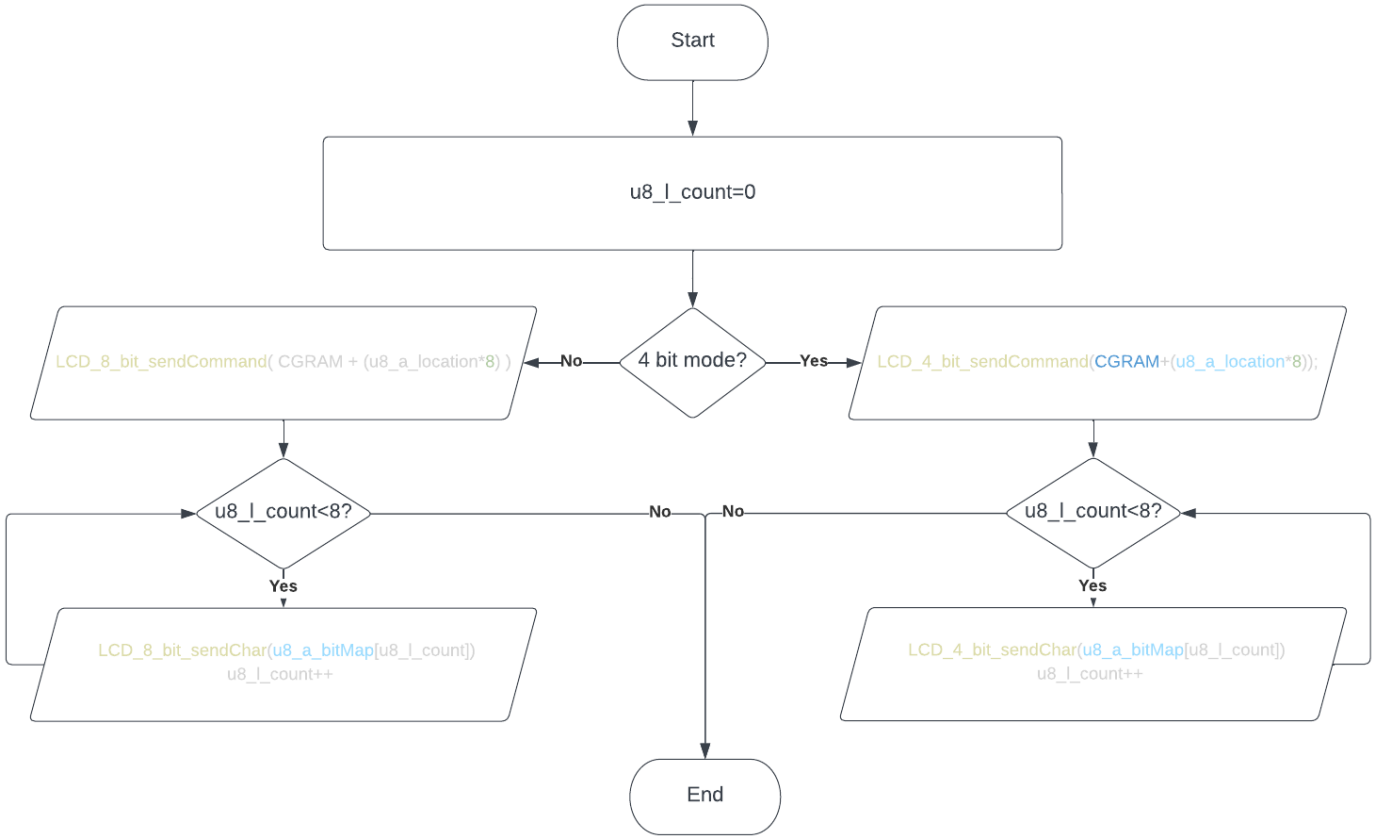
```
LCD_sendString_error LCD_sendString(uint8_t *u8_a_string);
```



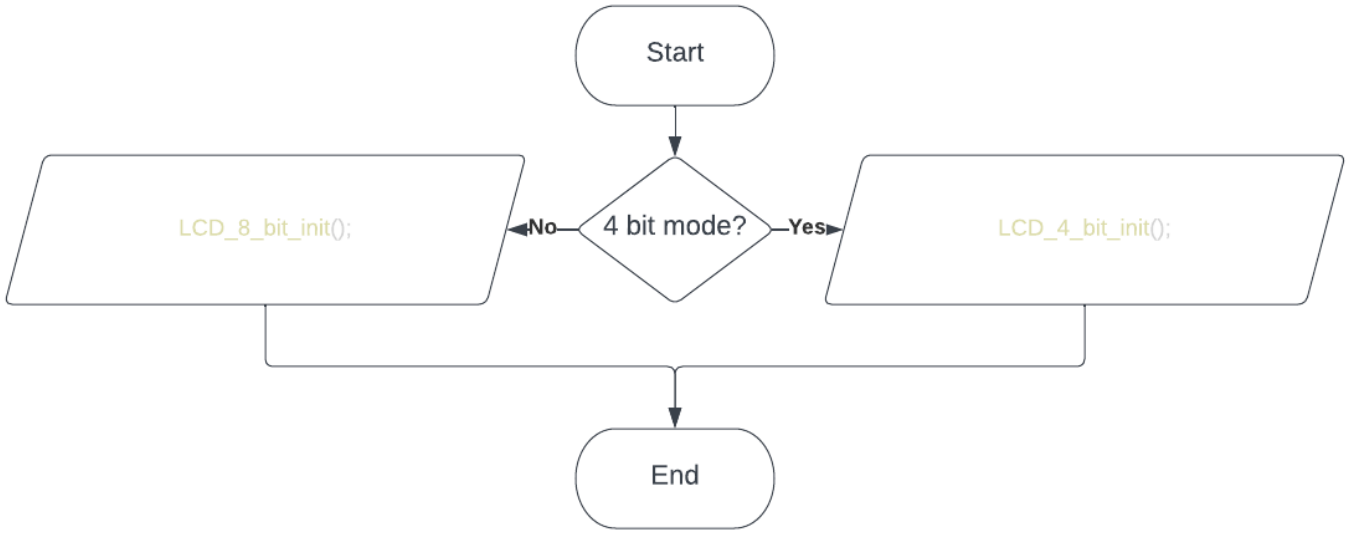
```
void LCD_goTo(uint8_t u8_a_row,uint8_t u8_a_column);
```



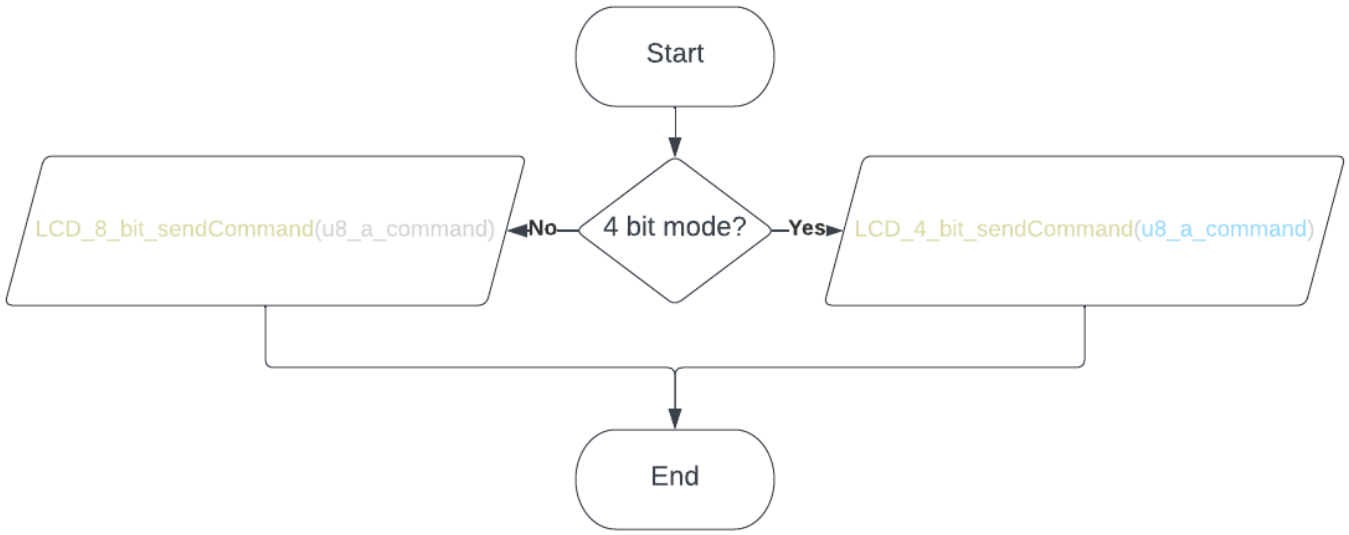
```
void LCD_createCustomCharacter(uint8_t *u8_a_bitMap,uint8_t u8_a_location);
```



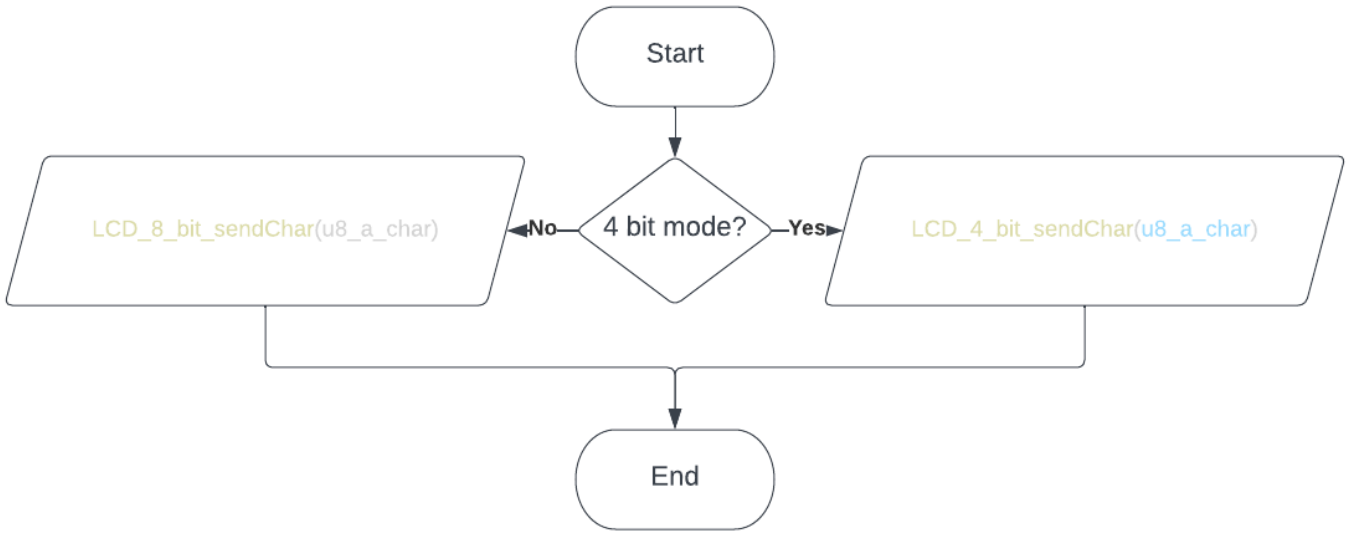
```
LCD_init_error LCD_init(void);
```



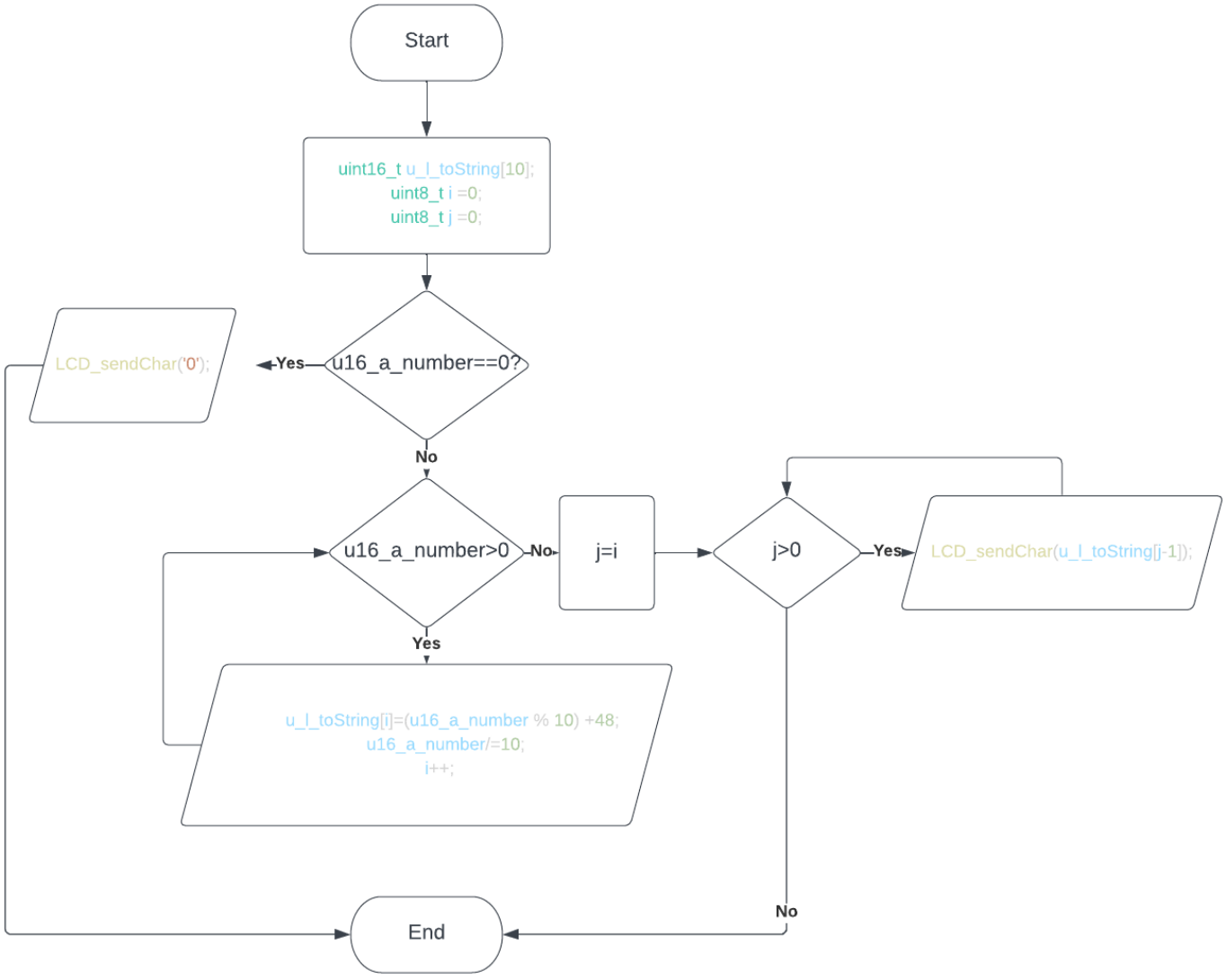
```
LCD_sendCommand_error LCD_sendCommand(uint8_t u8_a_command);
```



```
LCD_sendChar_error LCD_sendChar(uint8_t u8_a_char);
```

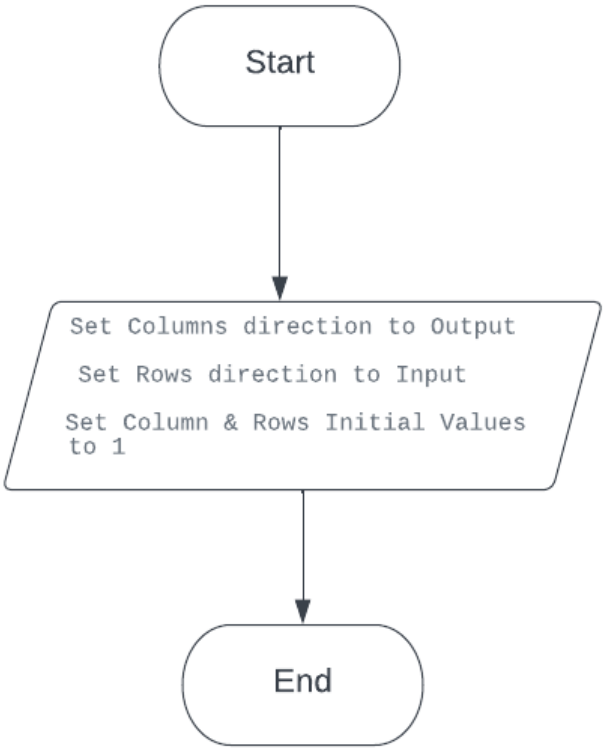


```
sendChar_error LCD_sendInteger(uint16_t u16_a_number);
```

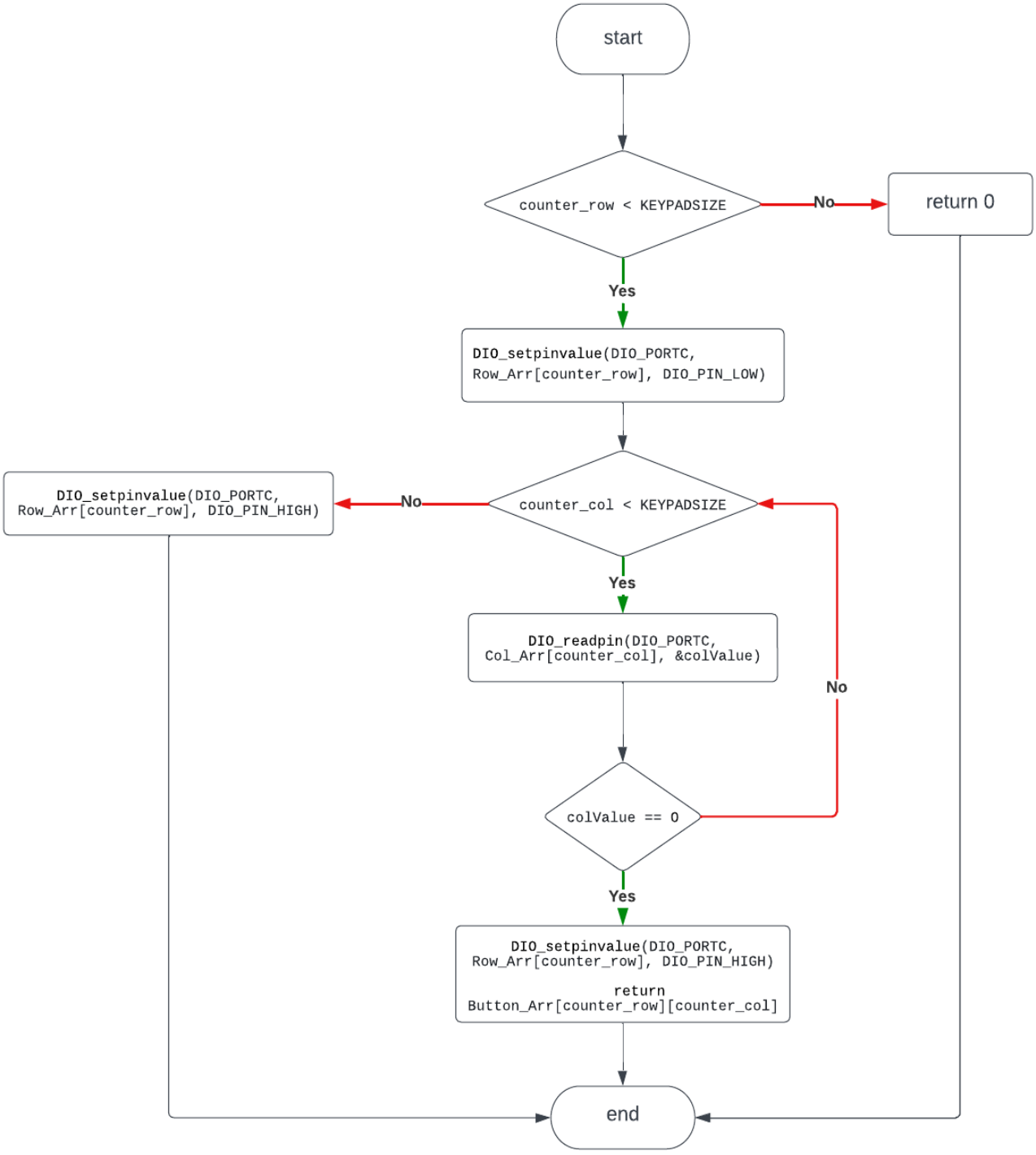


7.2 Keypad

```
void KEYPAD_init(void)
```

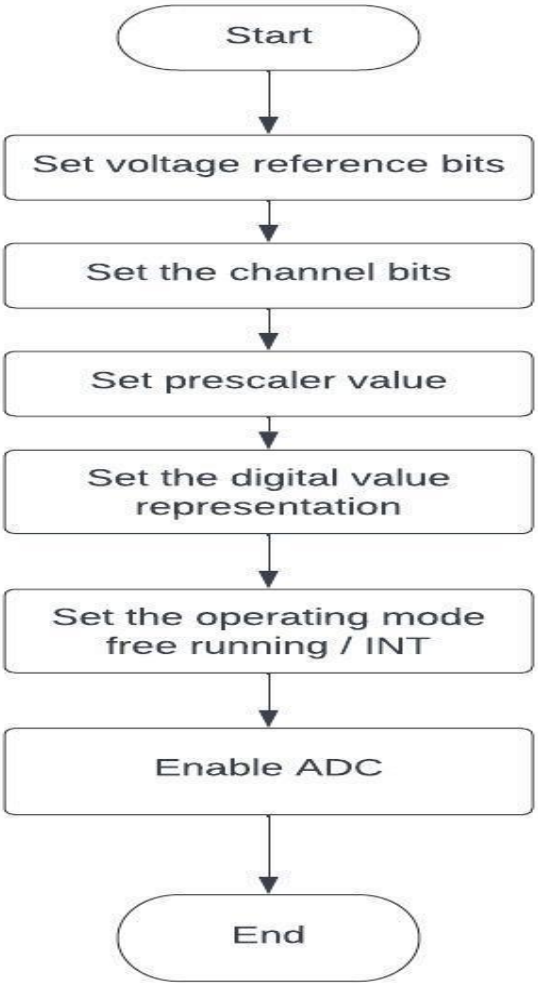


```
uint8_t KEYPAD_getpressedkey(void)
```

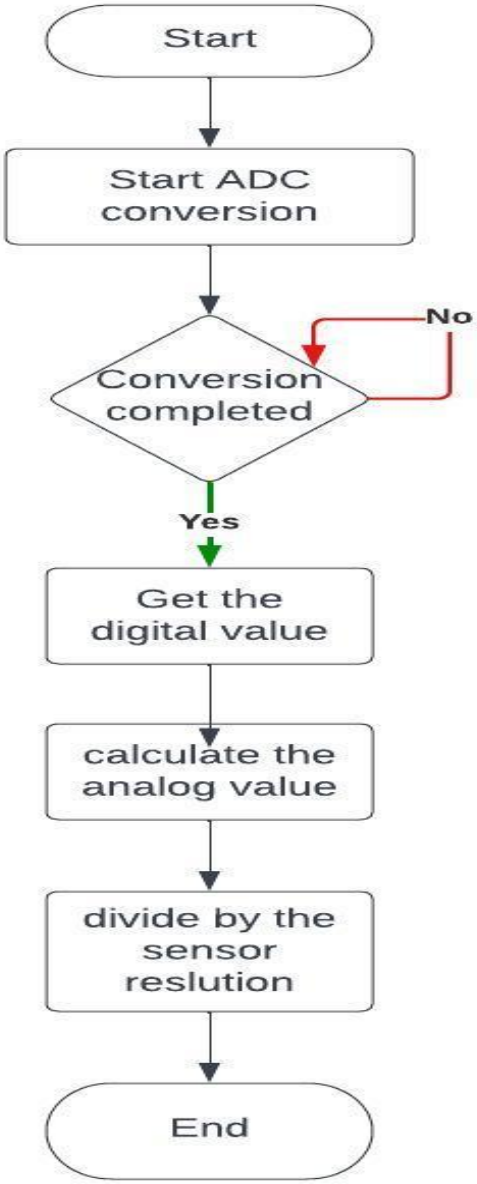


7.3 ADC

```
ADC_initstatus ADC_init(void);
```

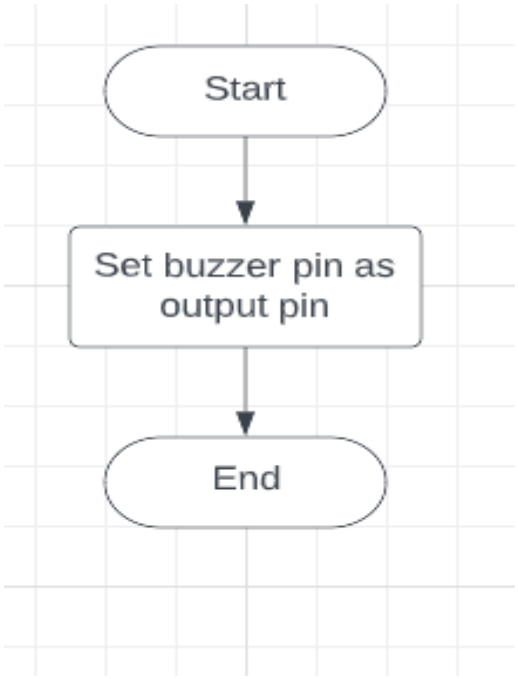


```c
uint32_t ADC_read(void);
```

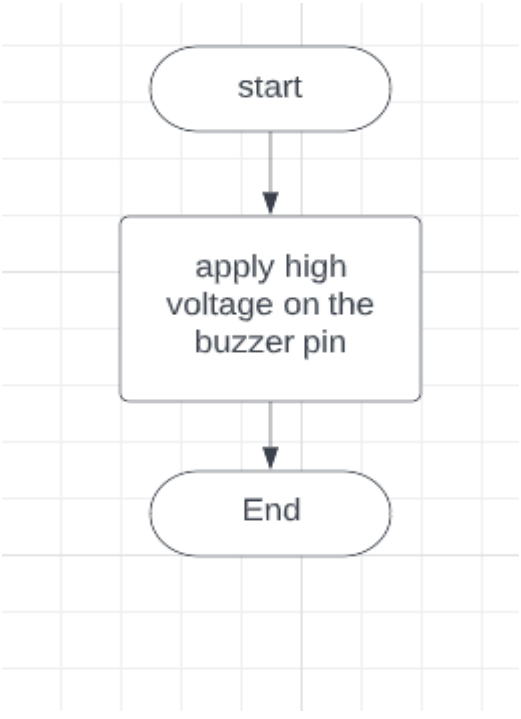


7.4 buzzer

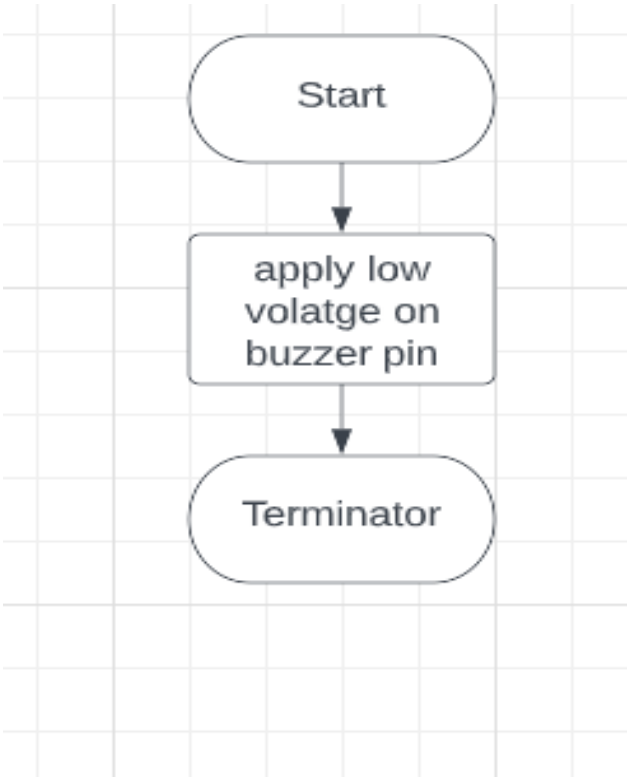
```
void BUZZ_init();
```




```
void BUZZ_on();
```

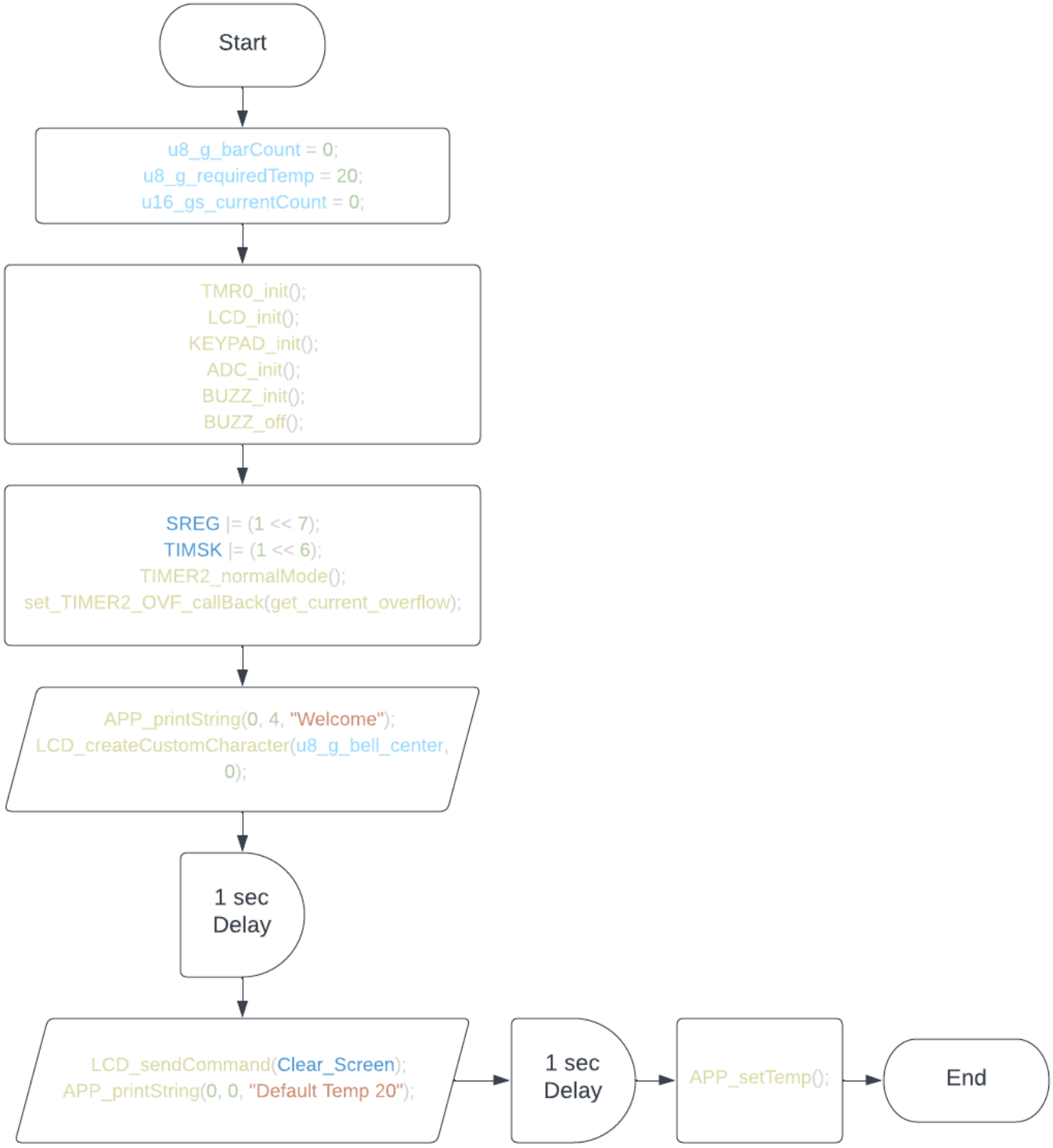


```
void BUZZ_off();
```

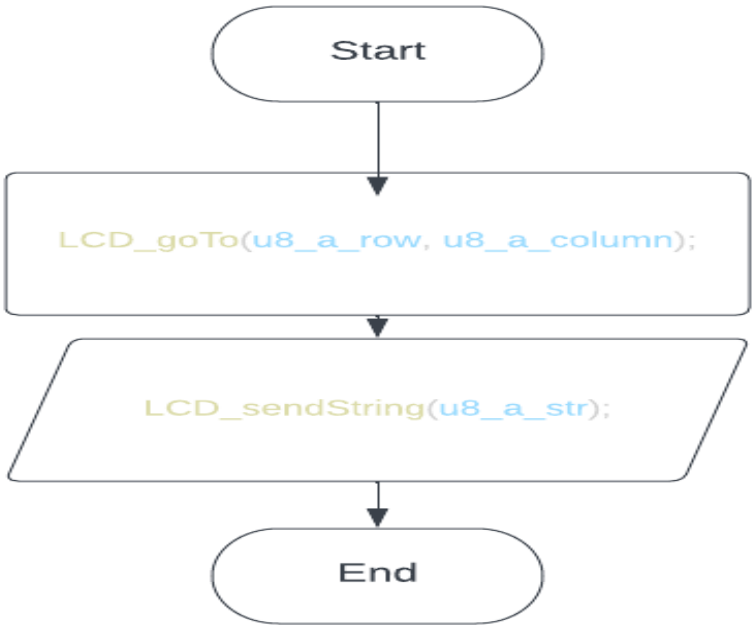


7.5 App

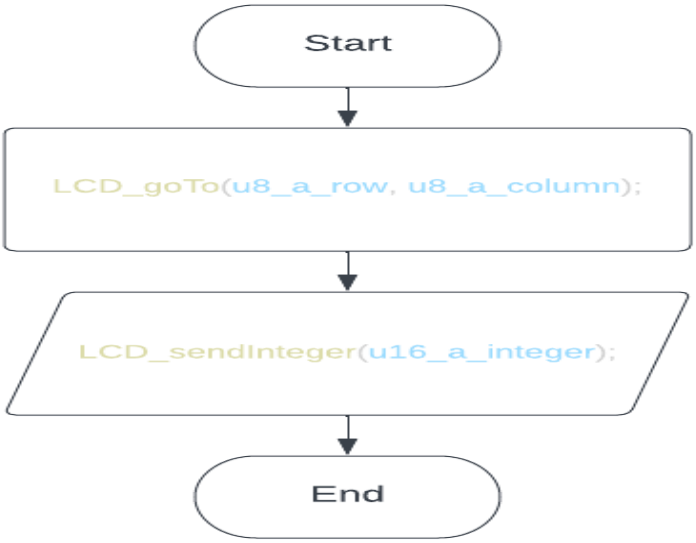
```
initError APP_init(void);
```



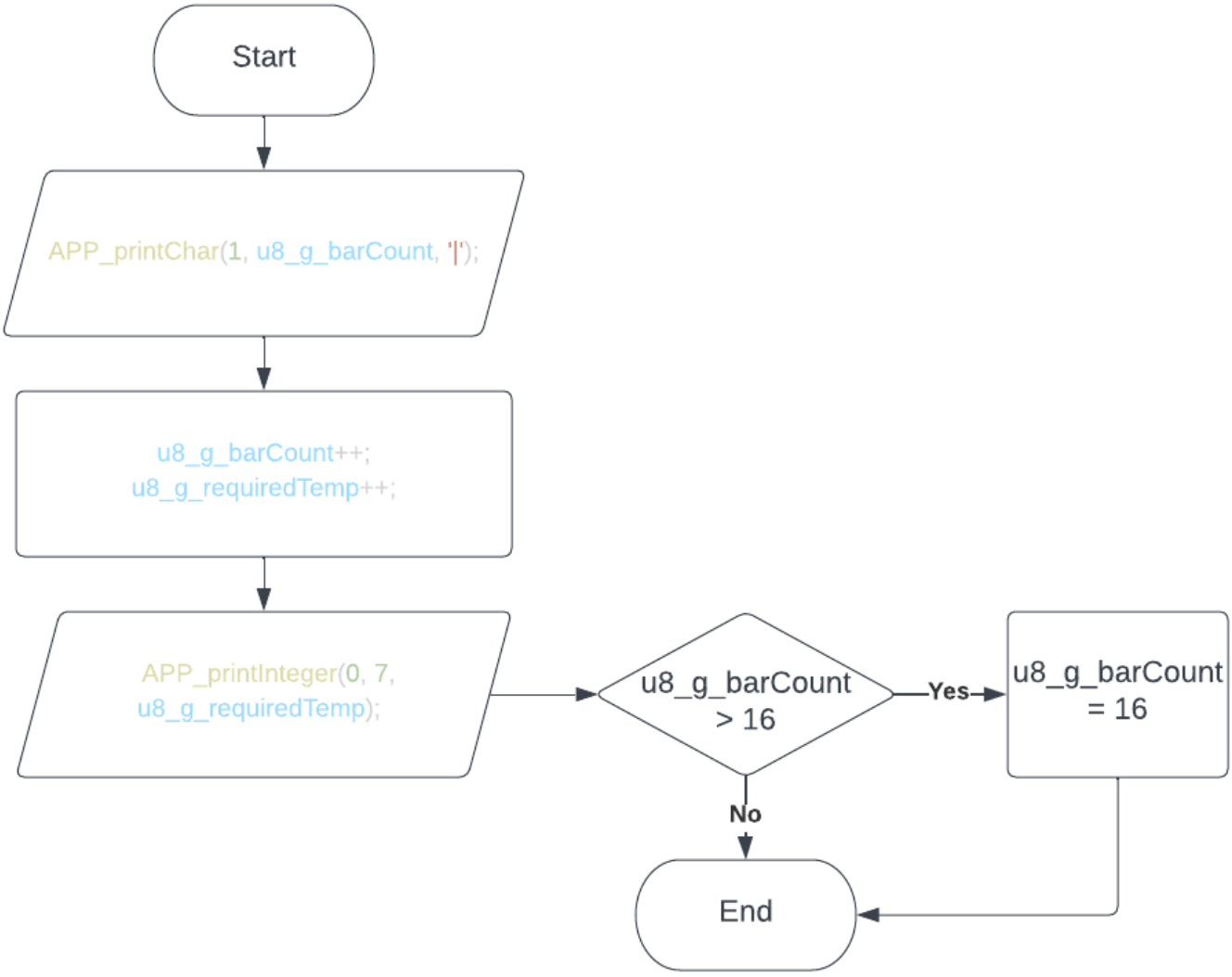
```
void APP_printString(uint8_t u8_a_row, uint8_t u8_a_column, uint8_t *u8_a_str);
```



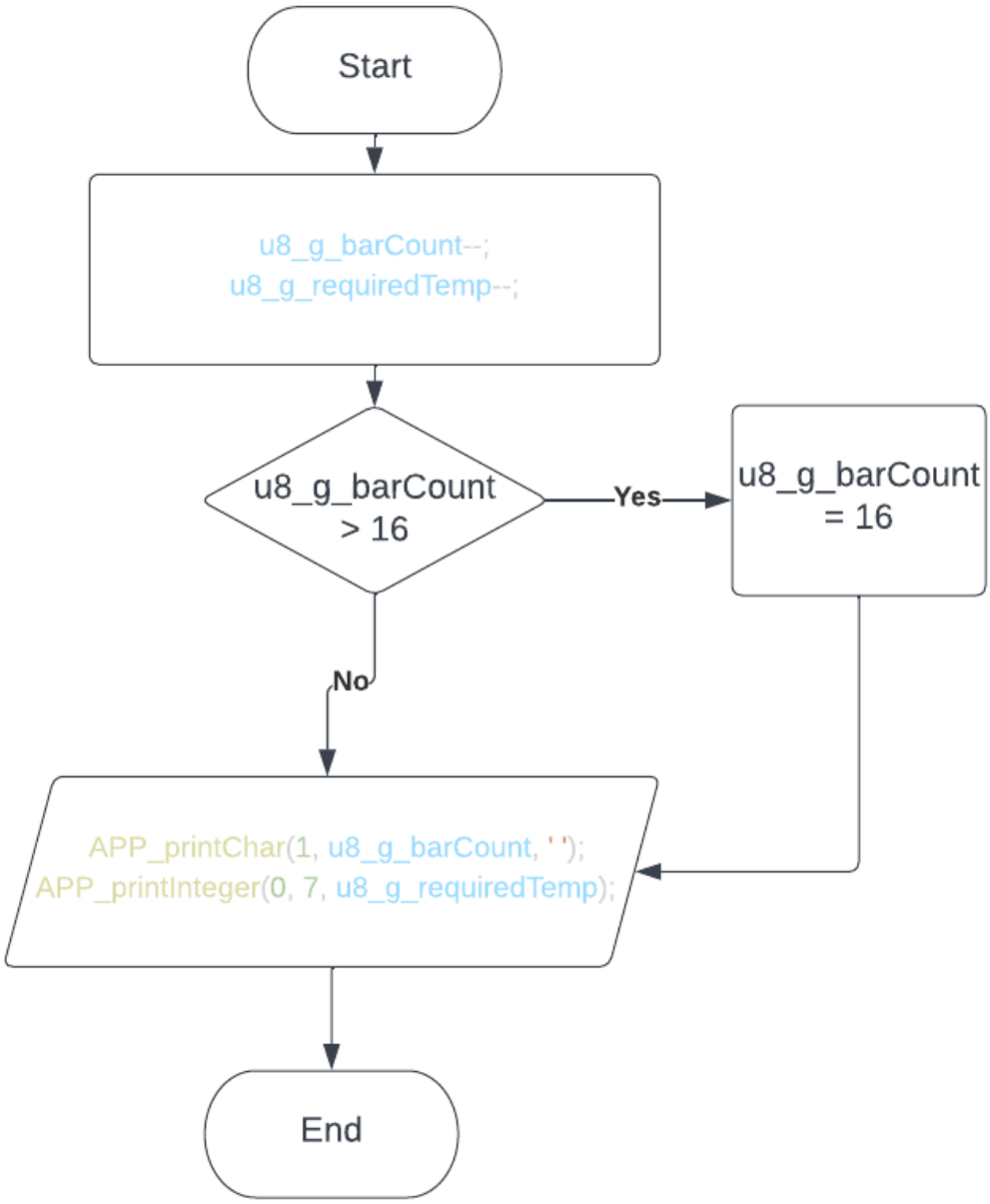

```
printInteger(uint8_t u8_a_row, uint8_t u8_a_column, uint16_t u16_a_integer);
```



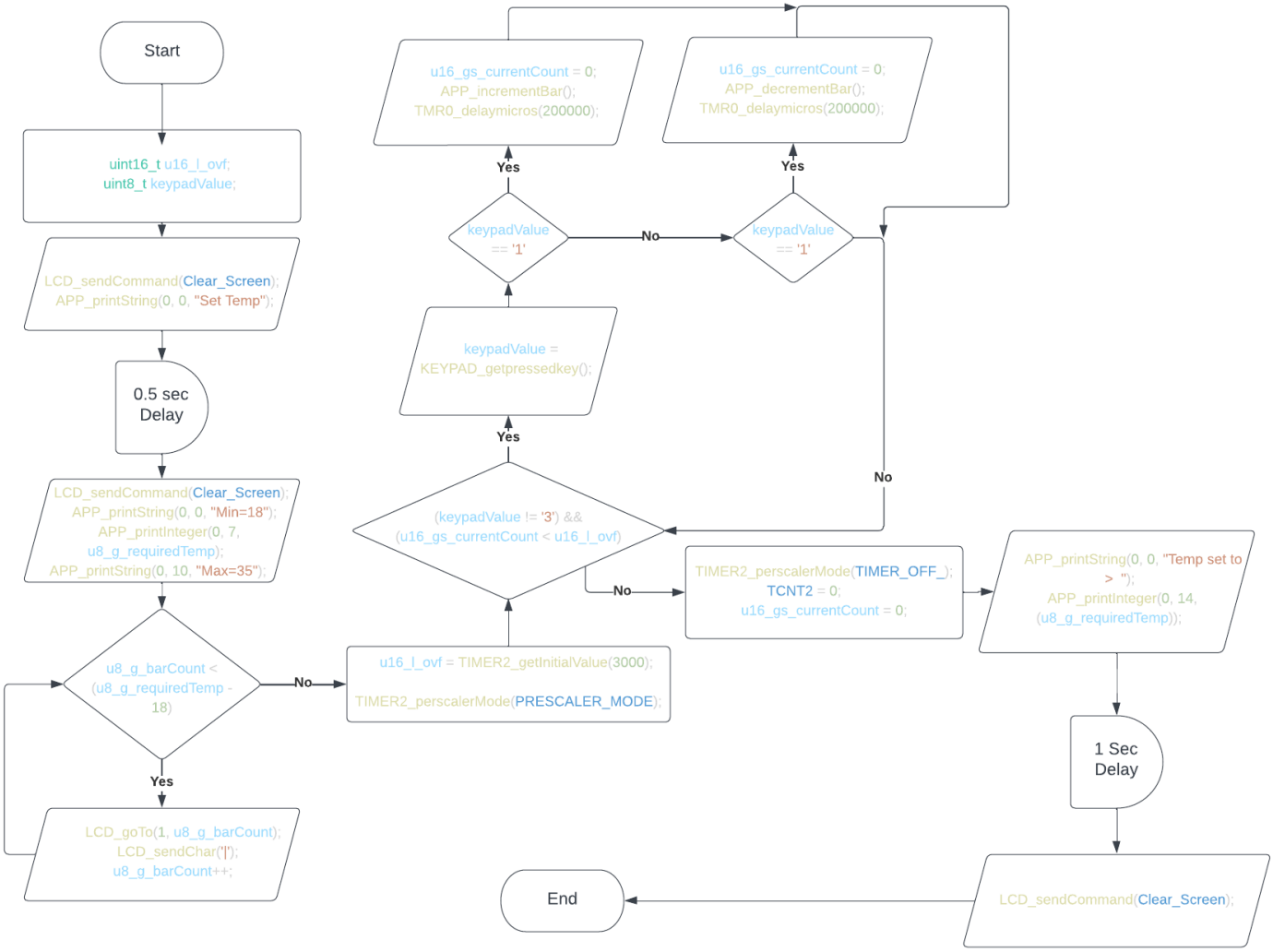
```
void APP_incrementBar();
```



```
void APP_decrementBar();
```



```
void APP_setTemp();
```



```
void APP_defaultView(void);
```