

# Air Conditioning System

## *Team 2*

*1- Mohab Ahmed*

*2- Anas Mahmoud*

*3- Mustafa Mohammed*

*4- Omar Taha*

## Table of Contents

1-Description .....	3
1.1 Hardware components.....	3
1.2 Software requirements .....	3
2-Layered architecture .....	4
3-System Flow Chart .....	5
4-Schematic Capture .....	6
5-Drivers Description .....	7
5.1 DIO Driver.....	7
5.2 Timer Driver .....	7
5.3 ADC Driver .....	7
5.4 Keypad Driver .....	7
5.5 LCD Driver.....	8
5.6 Application Driver .....	8
6-API's .....	9
6.1 DIO Driver.....	9
6.2 Timer0 Driver .....	9
6.3 Timer2 Driver .....	9
6.4 ADC Driver .....	10
6.5 Keypad Driver .....	10
6.6 LCD Driver.....	10
6.7 APP Driver .....	11
7- API's Flow Chart .....	12
7.1 LCD .....	12
7.2 Keypad.....	25
7.3 ADC.....	27
7.4 buzzer .....	29
7.5 Timer 2 .....	31
7.6 App .....	36

# 1-Description

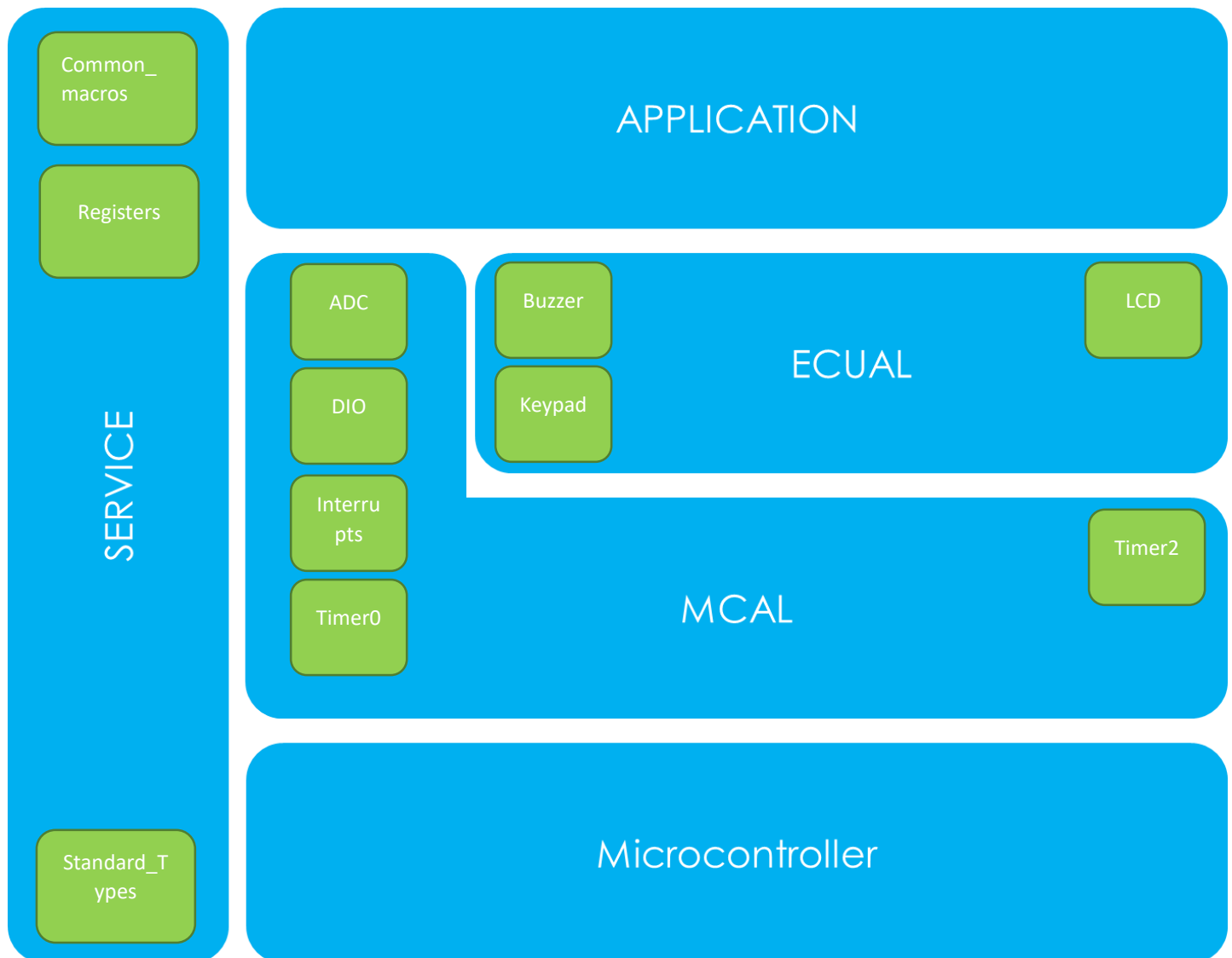
## 1.1 Hardware components

1. LCD (2\*16)
2. Keypad (3 \* 3) (Note: 4 buttons will be used)
3. Temperature sensor (LM35)
4. Buzzer

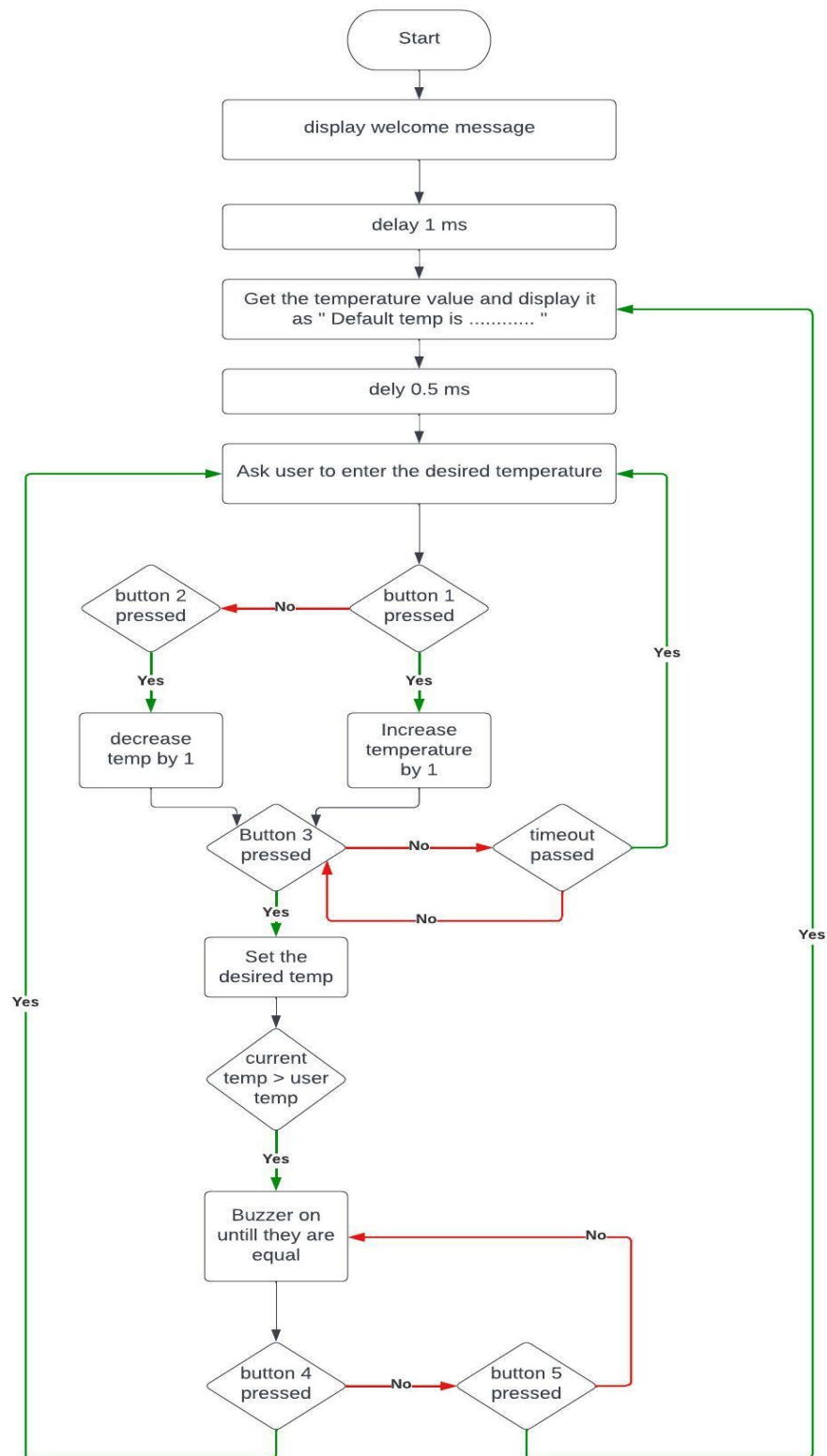
## 1.2 Software requirements

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

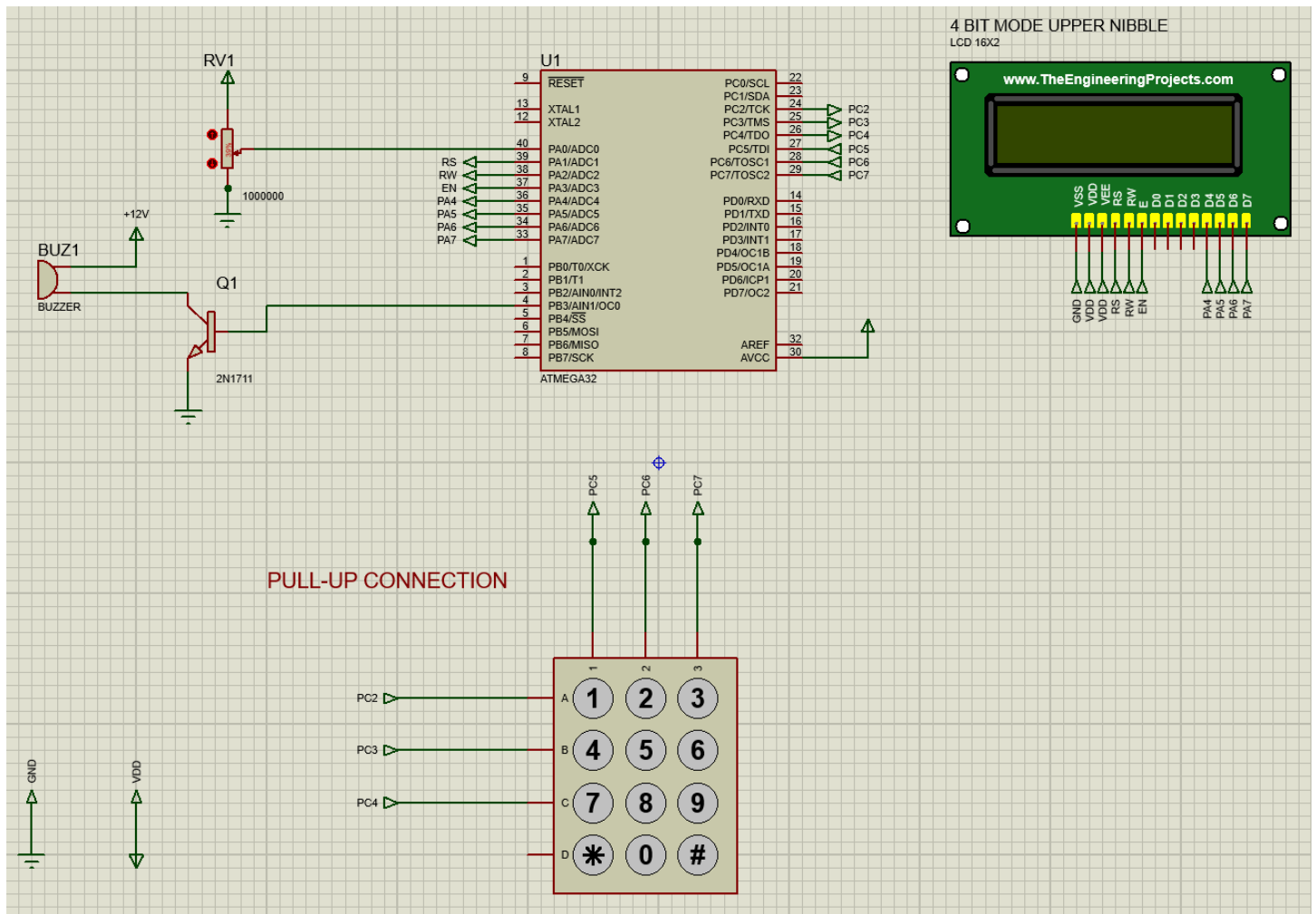
## 2-Layered architecture



### 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 write 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\_delaymicros(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);

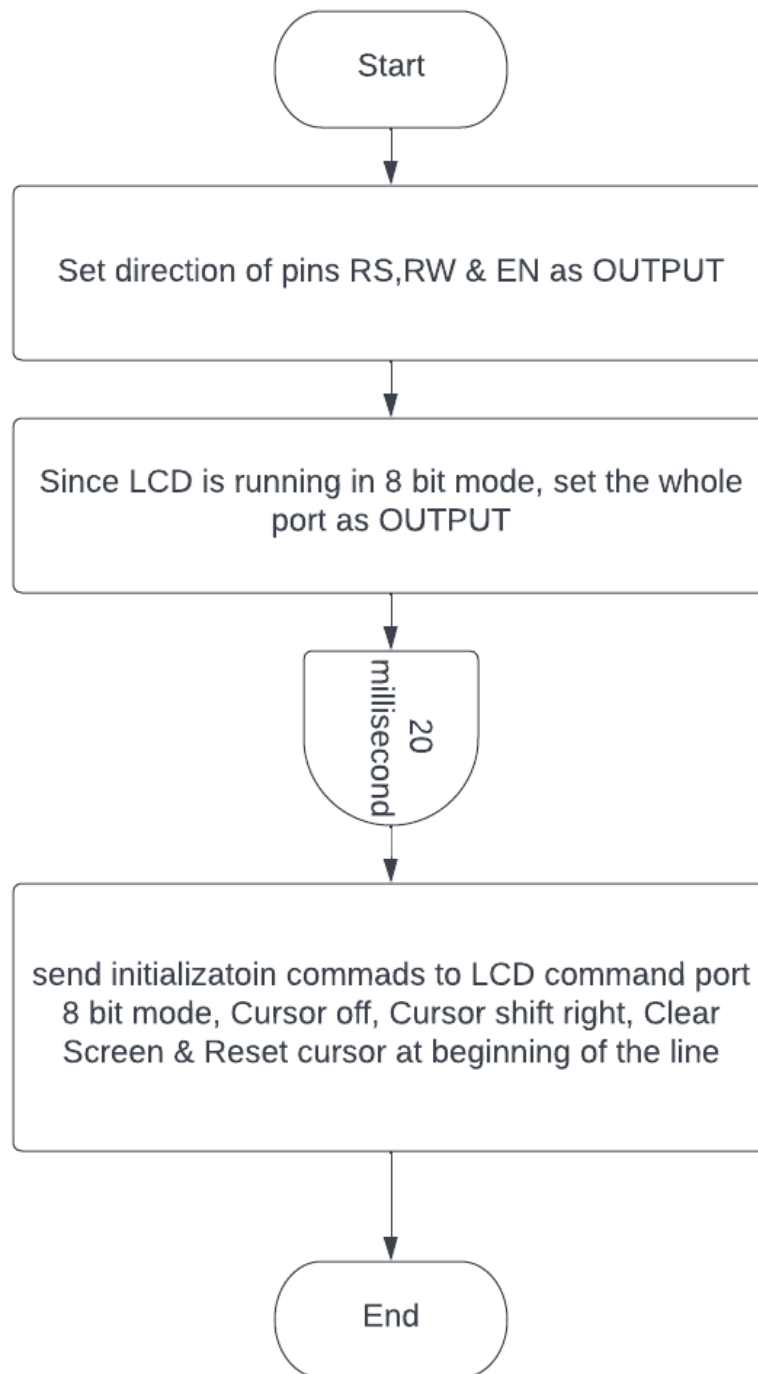
## 6.7 APP Driver

- 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);

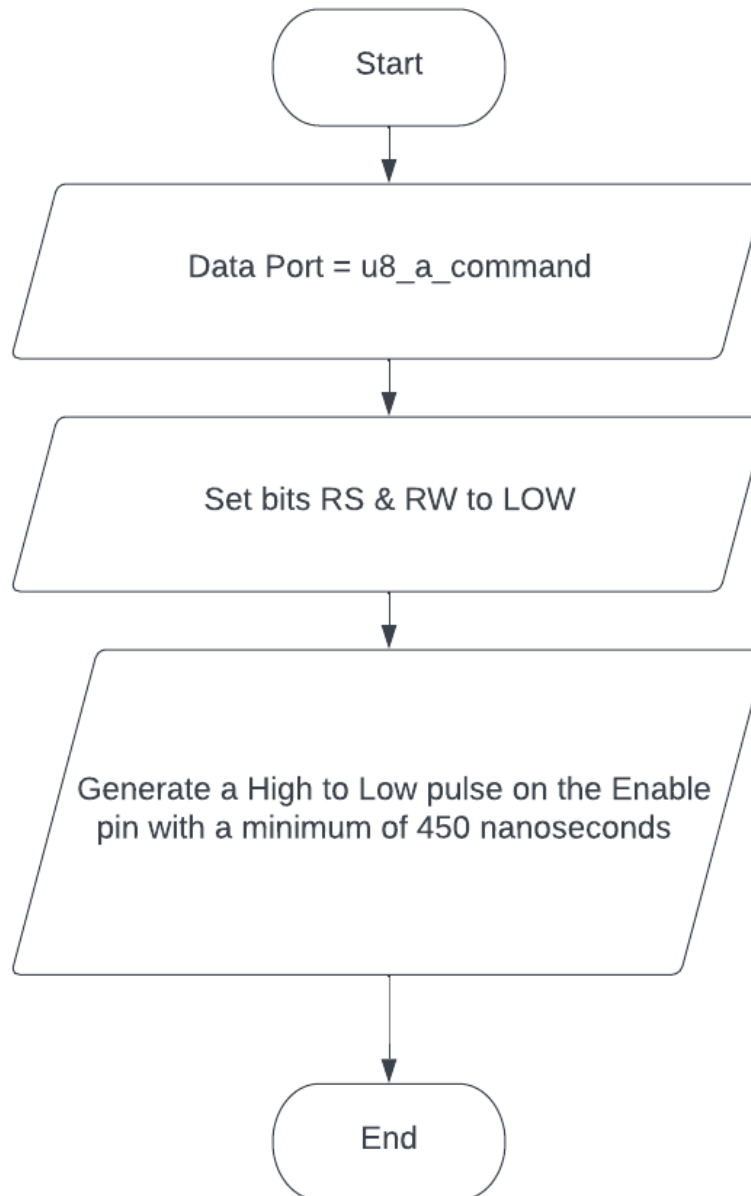
# 7- API's Flow Chart

## 7.1 LCD

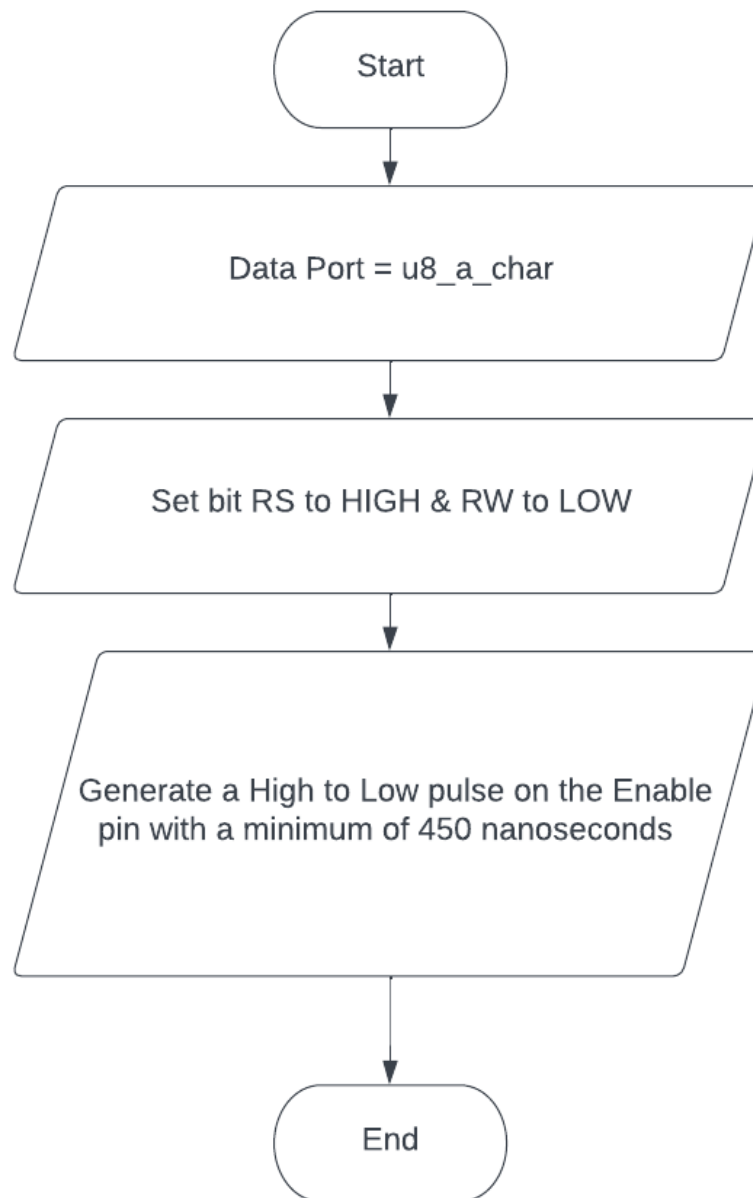
```
LCD_init_error LCD_8_bit_init(void);
```



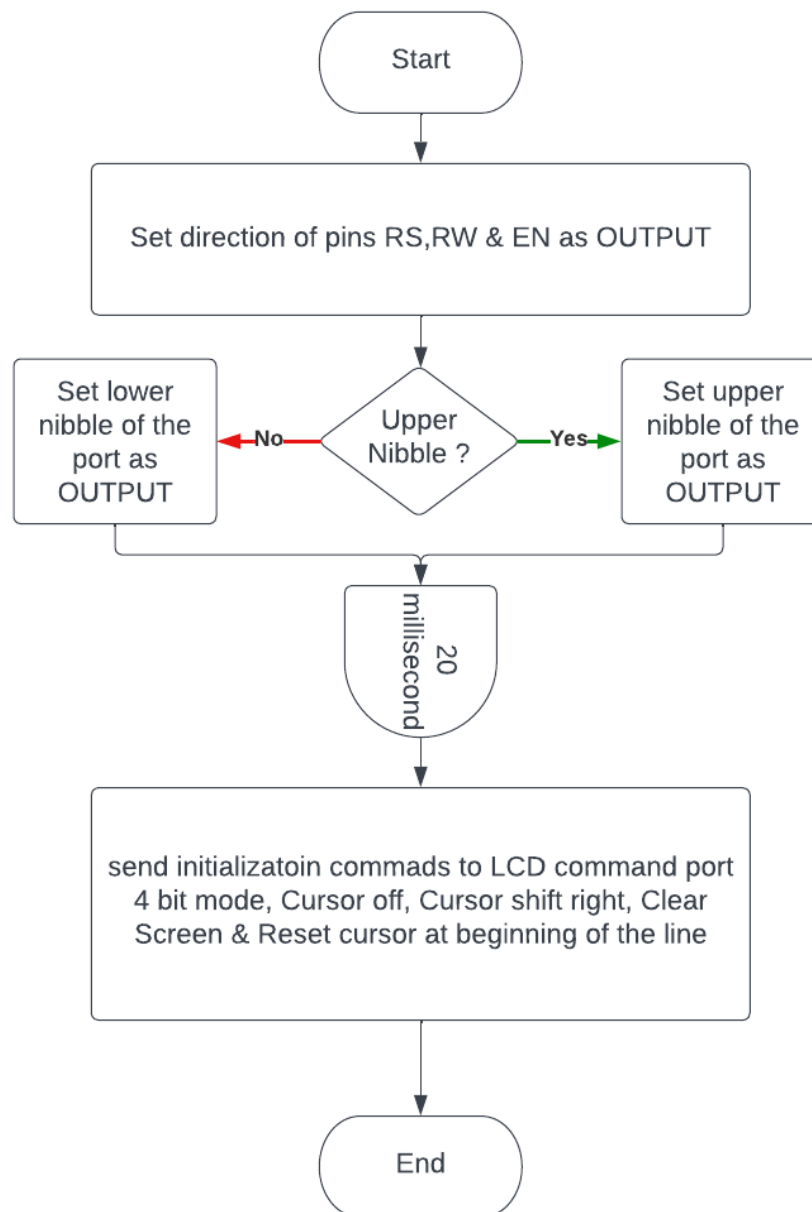
```
LCD_sendCommand_error LCD_8_bit_sendCommand(uint8_t u8_a_command);
```



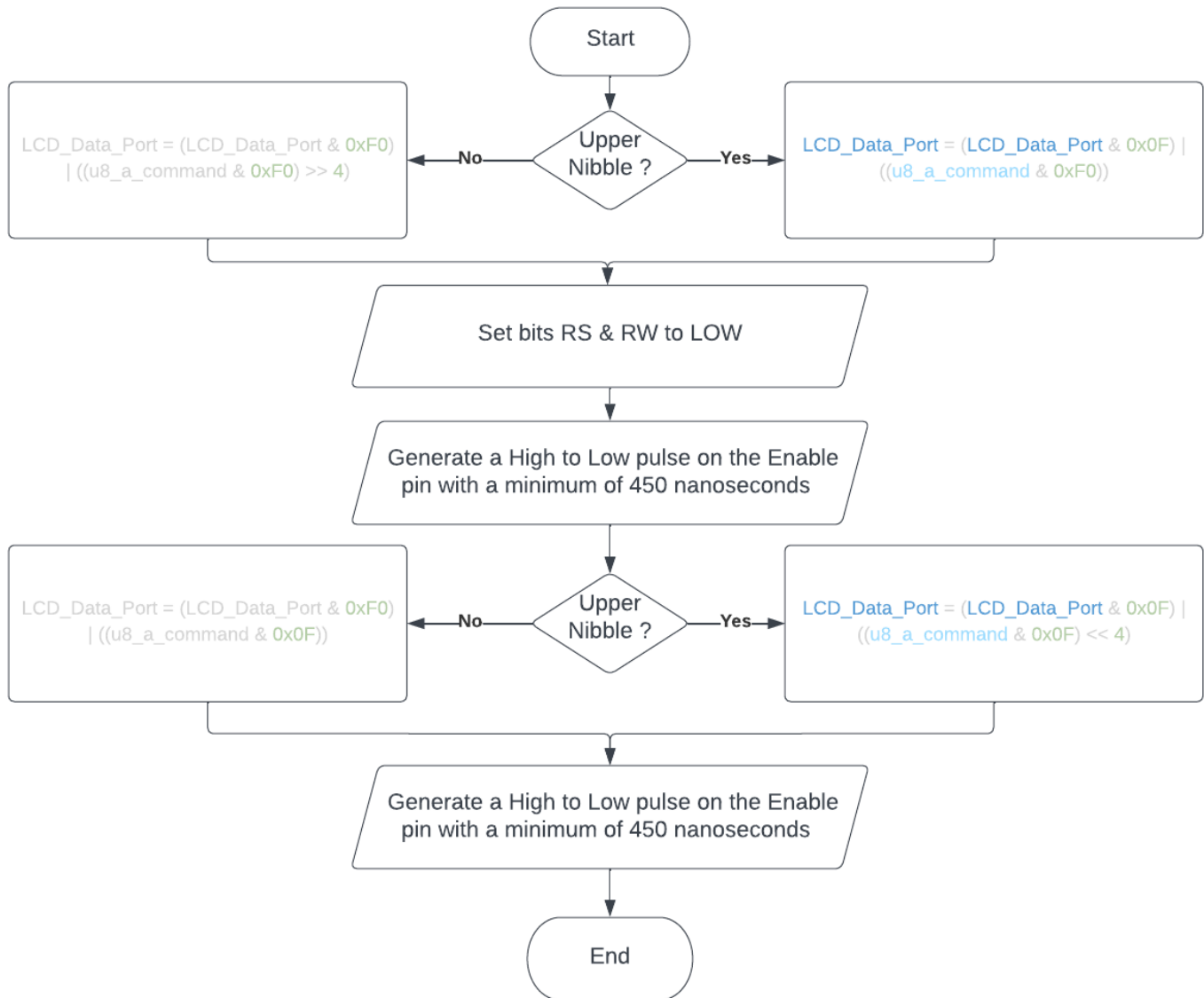
```
LCD_sendChar_error LCD_8_bit_sendChar(uint8_t u8_a_char);
```



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

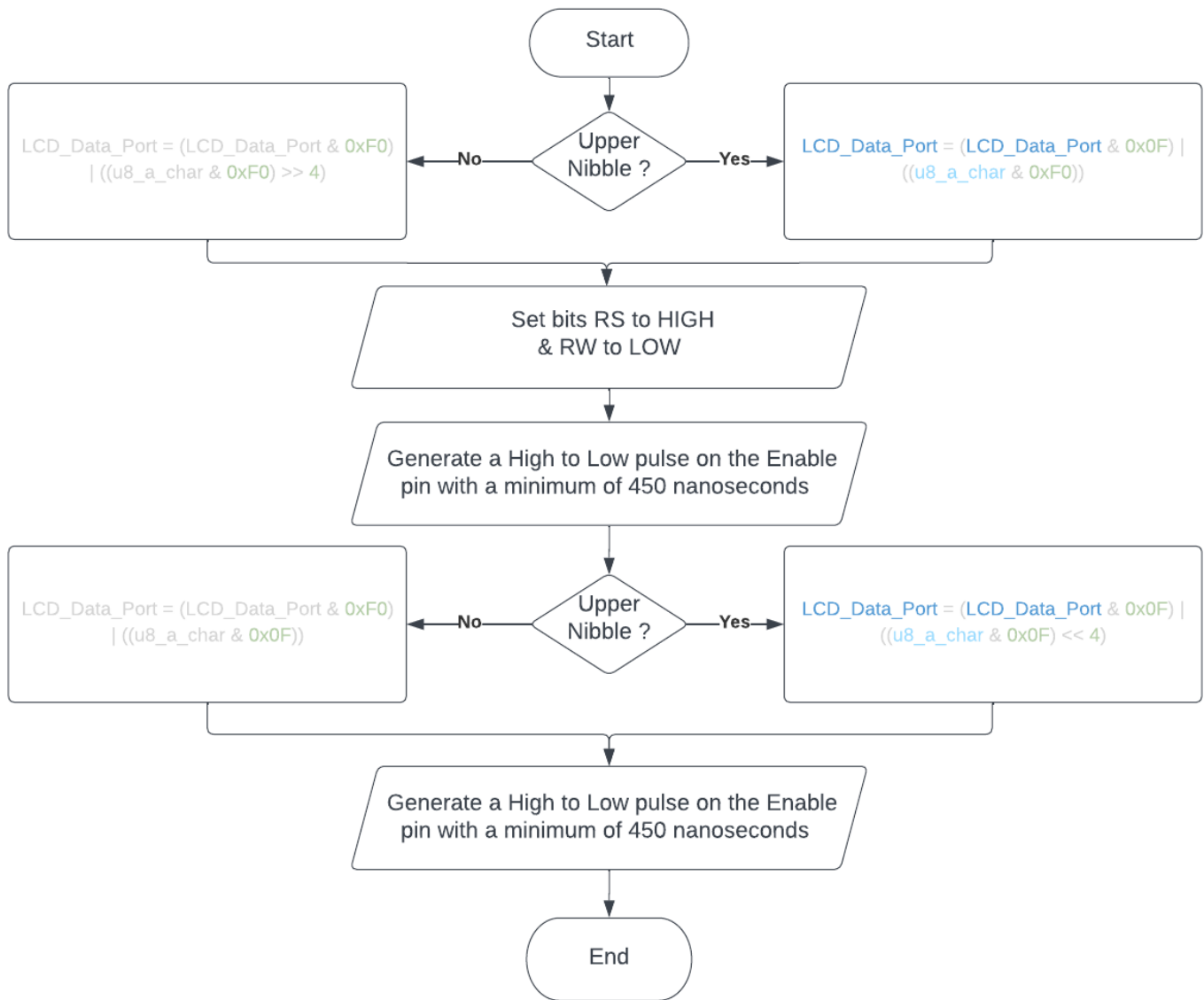


```
LCD_sendCommand_error LCD_4_bit_sendCommand(uint8_t u8_a_command);
```

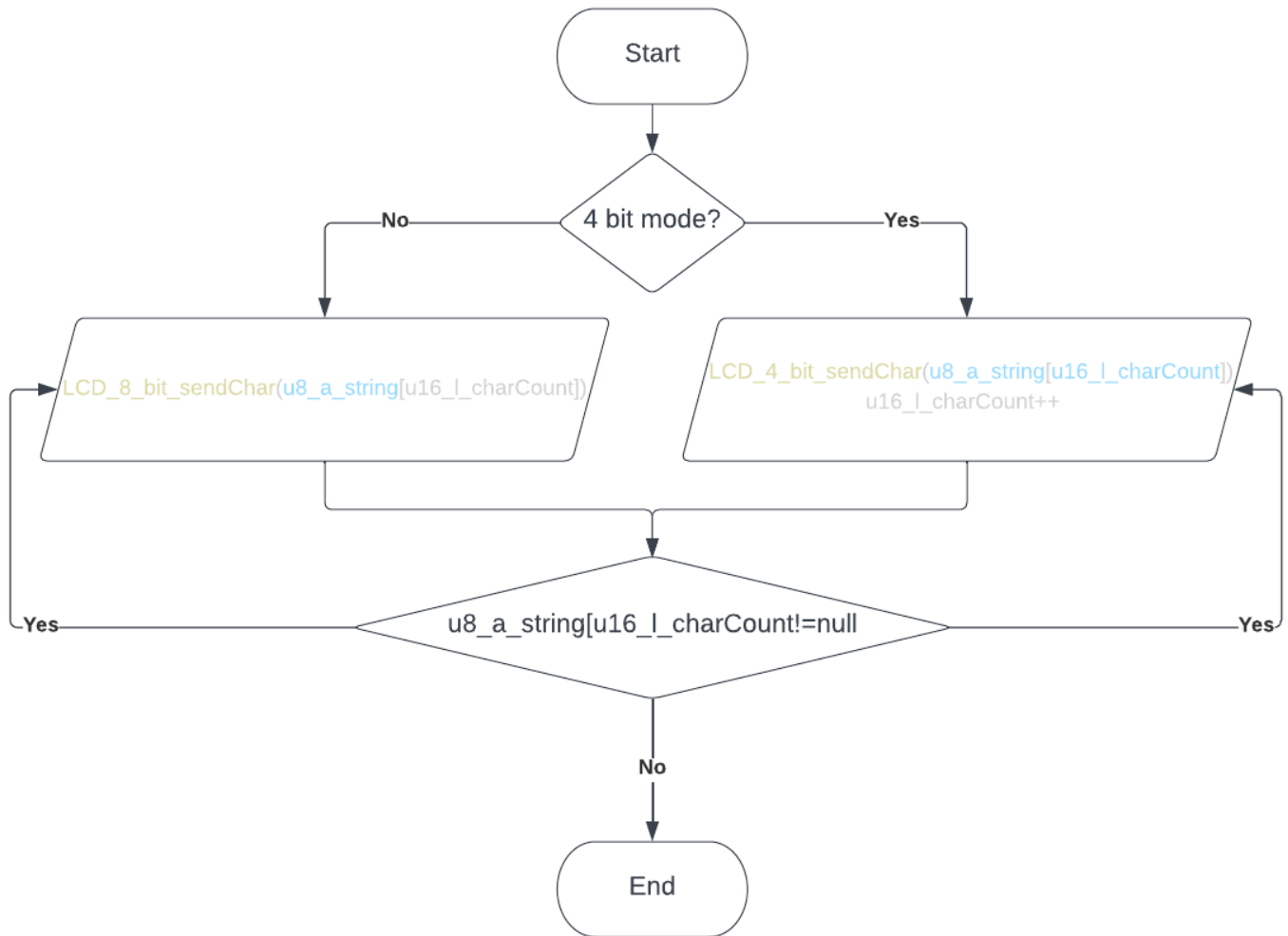




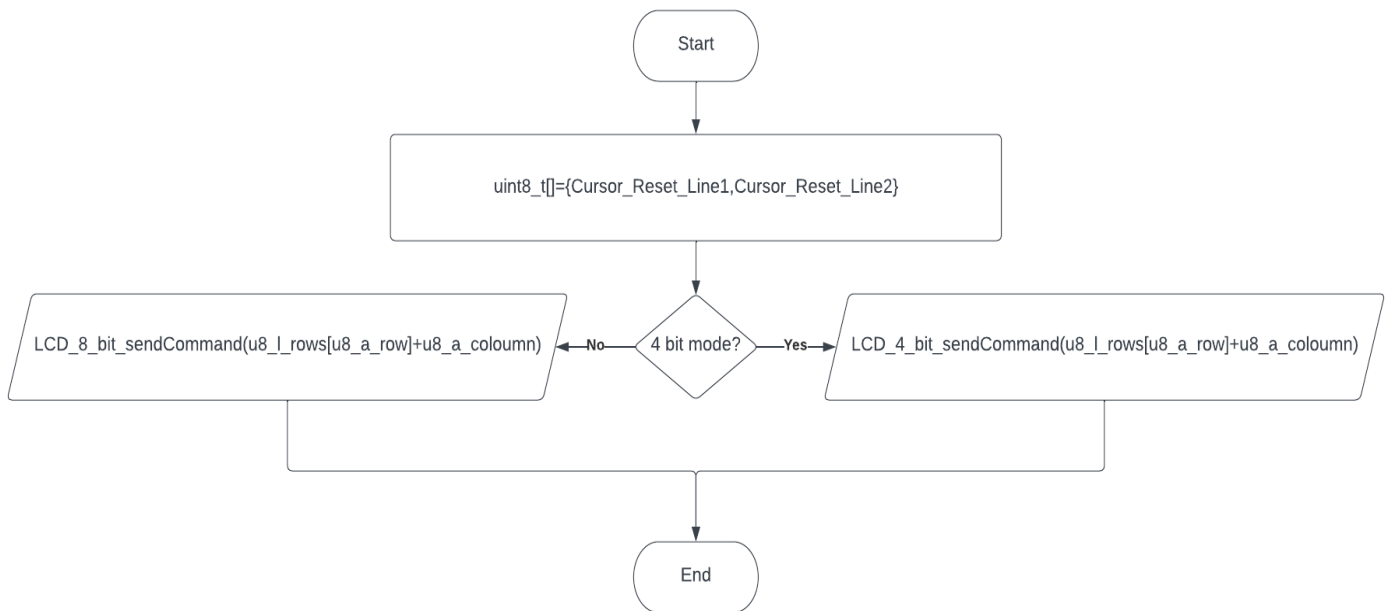
```
LCD_sendChar_error LCD_4_bit_sendChar(uint8_t u8_a_char);
```



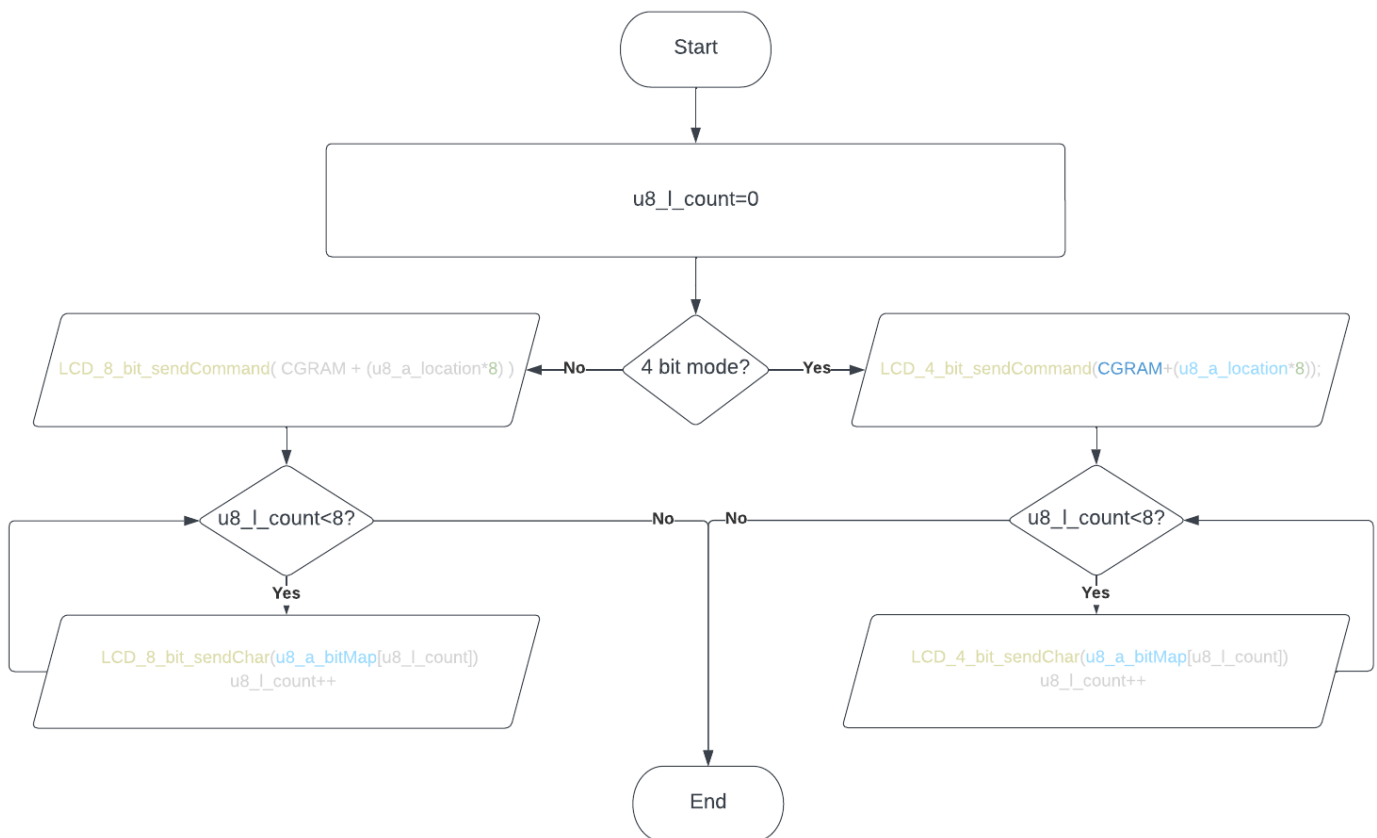
```
LCD_sendString_error LCD_sendString(uint8_t *u8_a_string);
```



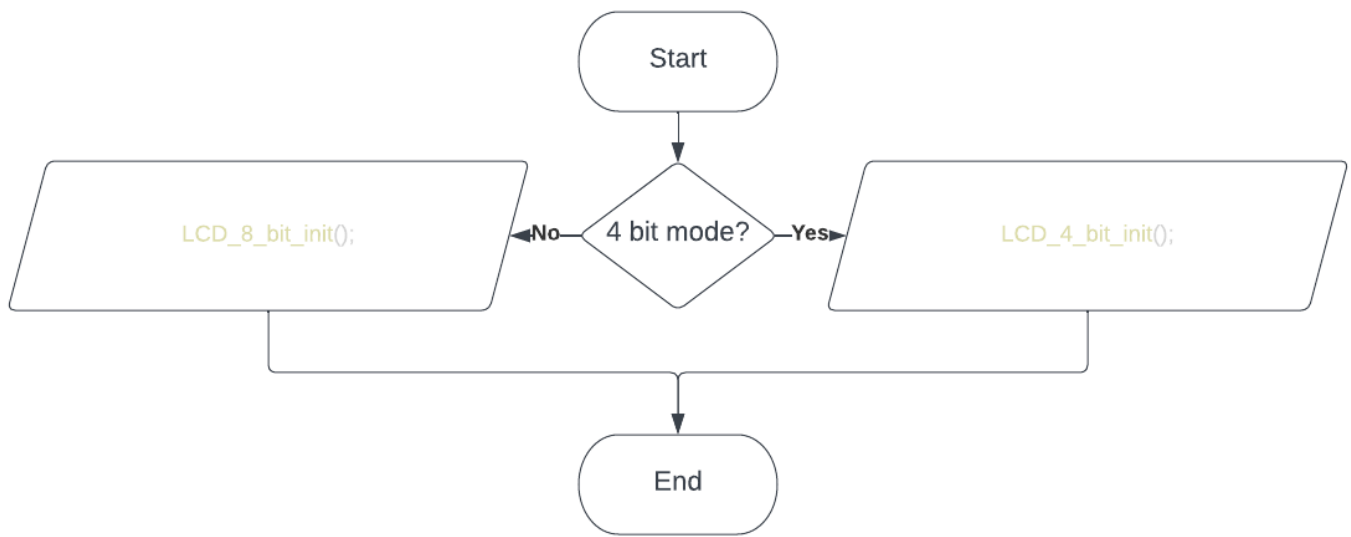
```
void LCD_goto(uint8_t u8_a_row,uint8_t u8_a_column);
```



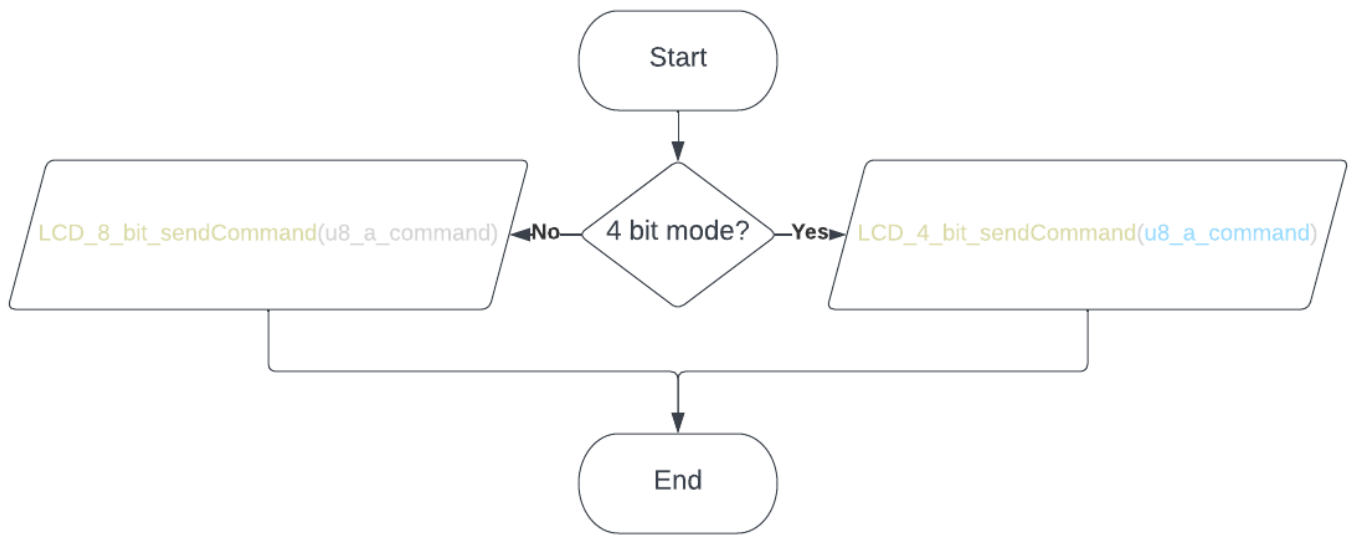
```
void LCD_createCustomCharacter(uint8_t *u8_a_bitMap,uint8_t u8_a_location);
```



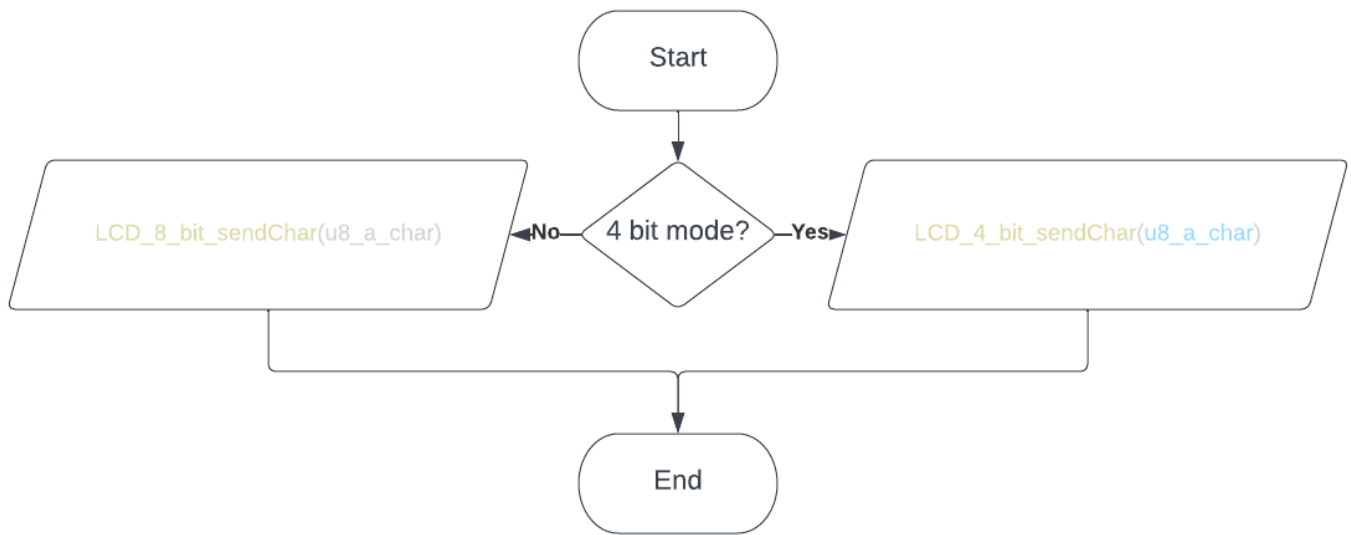
```
LCD_init_error LCD_init(void);
```



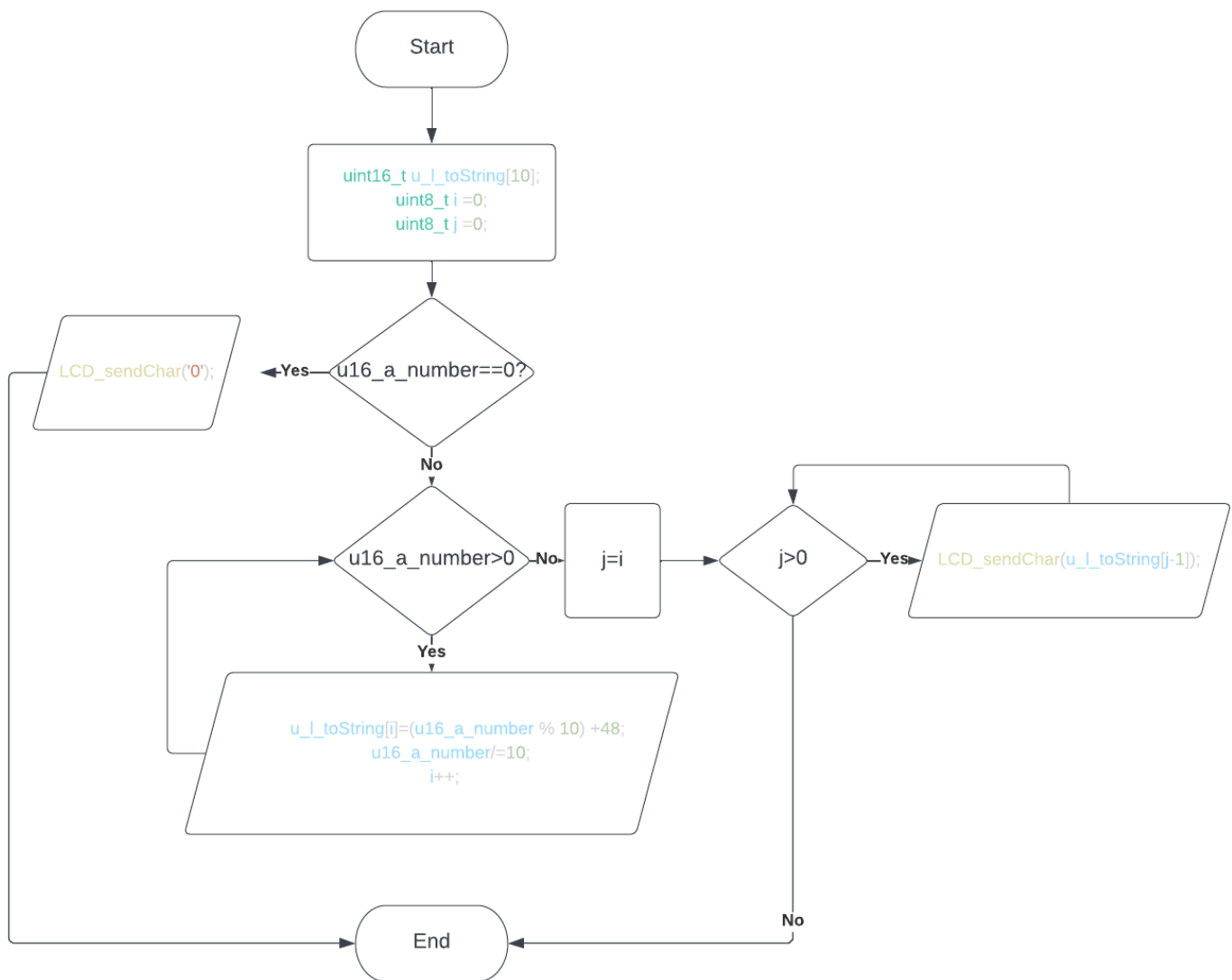
```
LCD_sendCommand_error LCD_sendCommand(uint8_t u8_a_command);
```



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



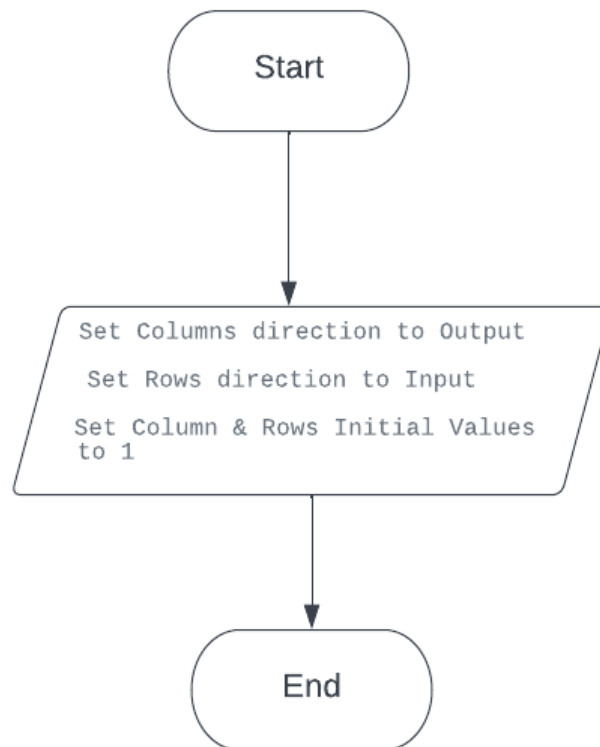
```
sendChar_error LCD_sendInteger(uint16_t u16_a_number);
```



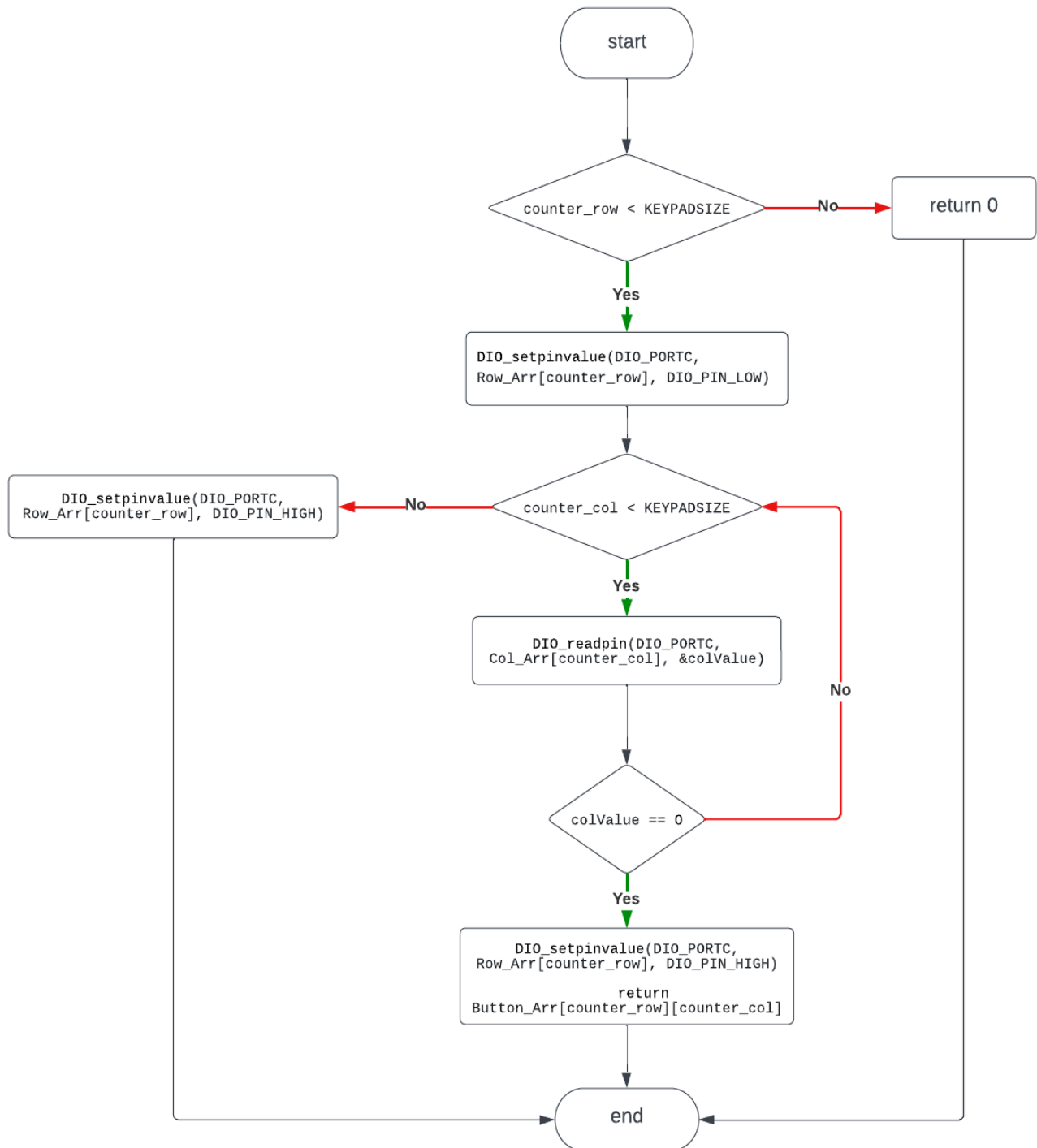


## 7.2 Keypad

```
void KEYPAD_init(void)
```

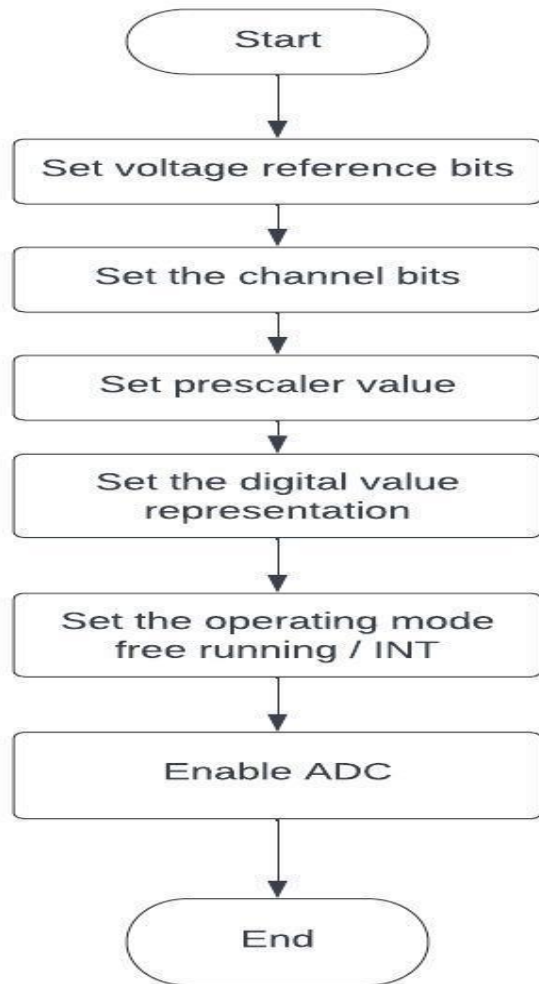


```
uint8_t KEYPAD_getpressedkey(void)
```

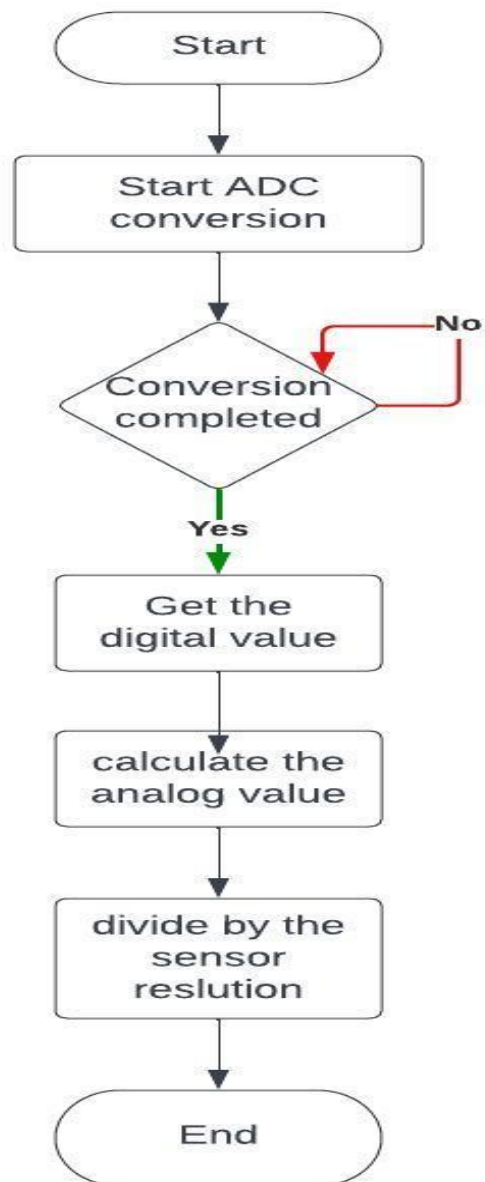


## 7.3 ADC

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

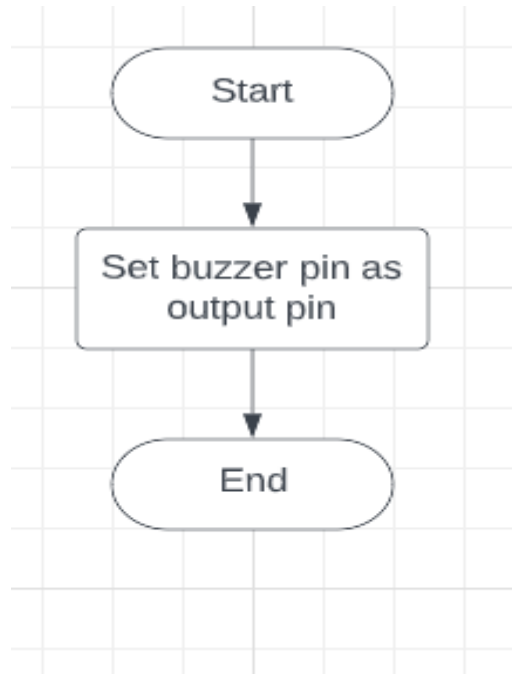


```
uint32_t ADC_read(void);
```

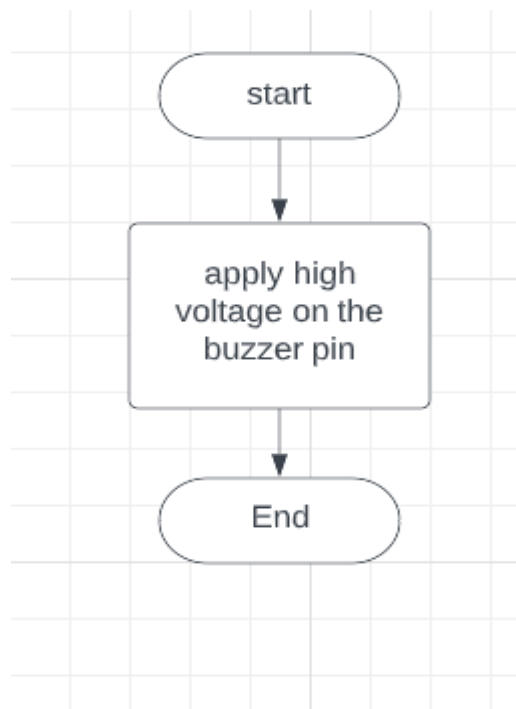


## 7.4 buzzer

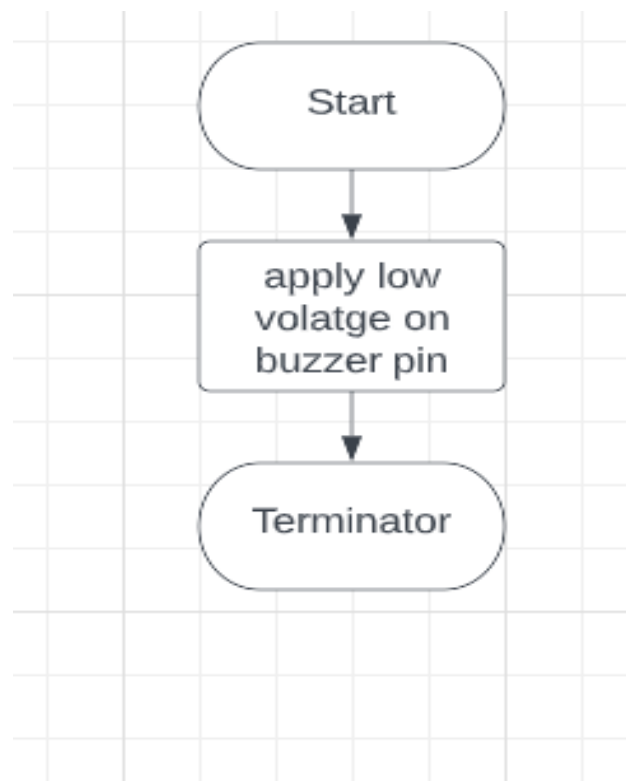
```
void BUZZ_init();
```



```
void BUZZ_on();
```

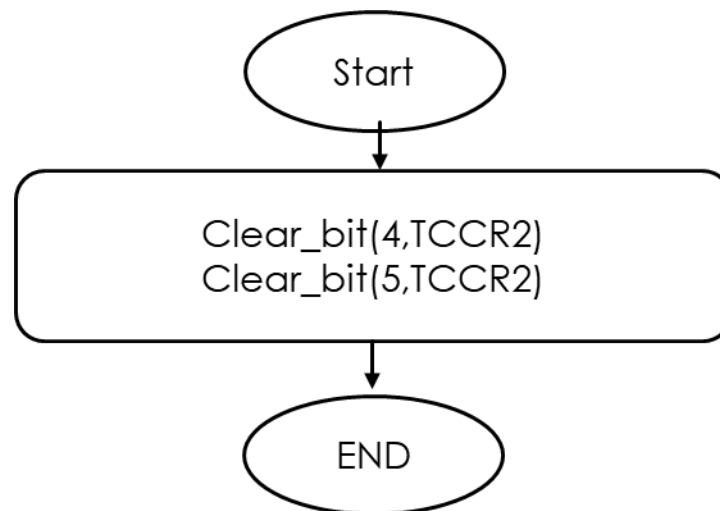


```
void BUZZ_off();
```

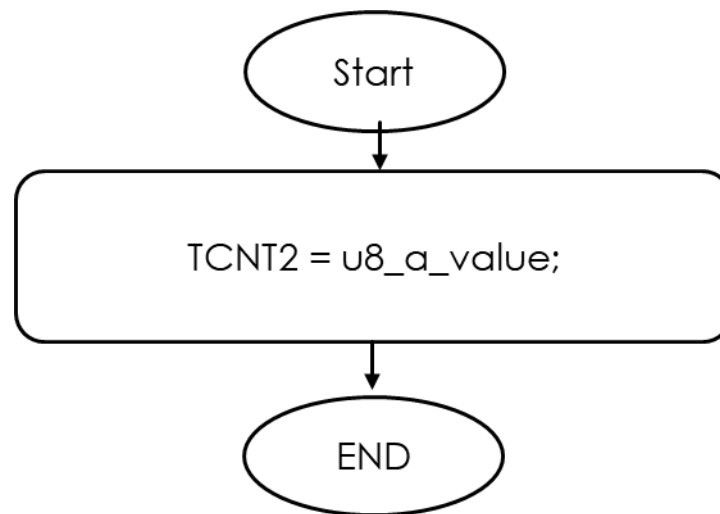


## 7.5 Timer 2

```
err_state TIMER2_normalMode(void);
```

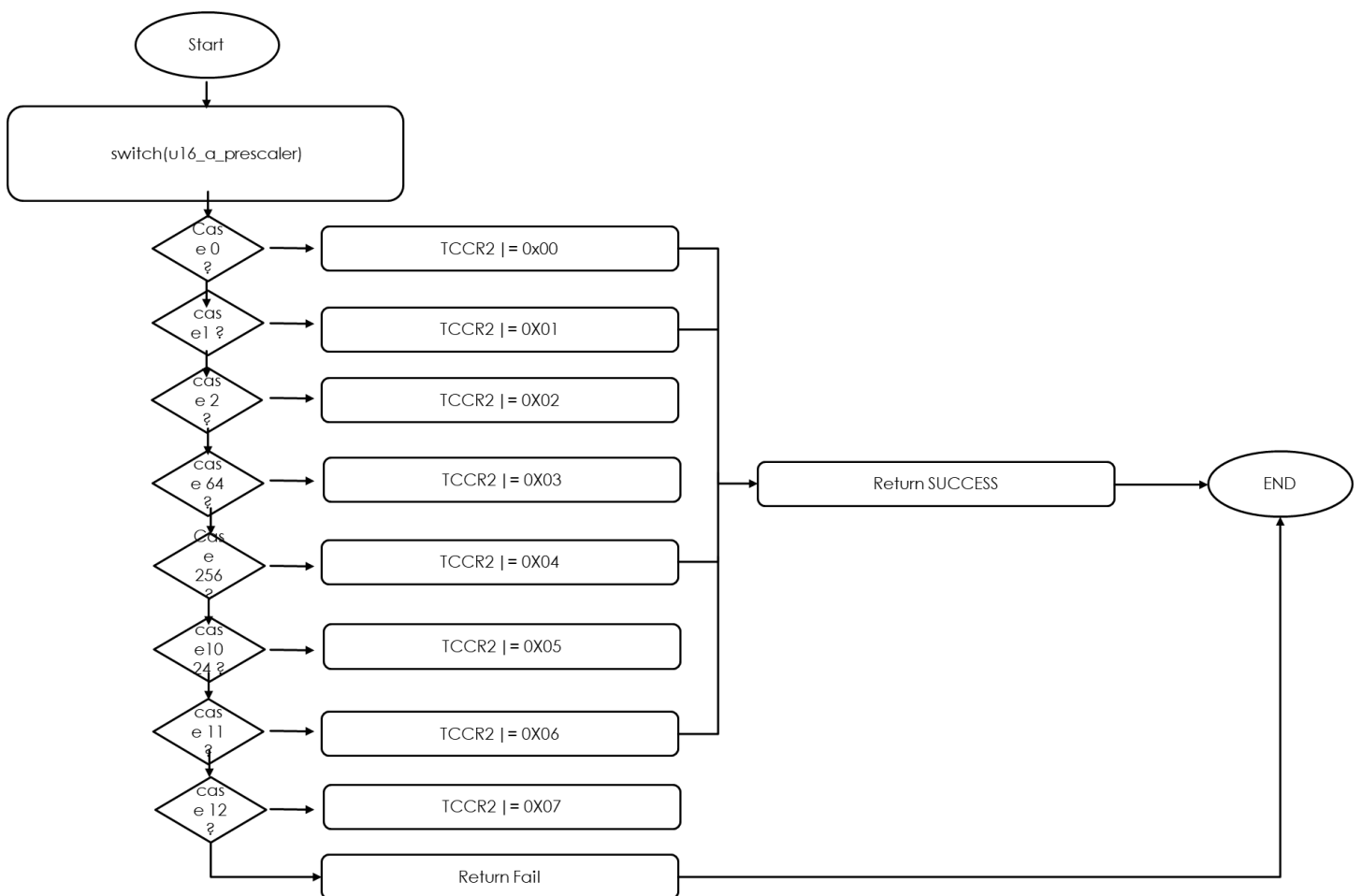


```
err_state TIMER2_initialValue(uint8_t value);
```

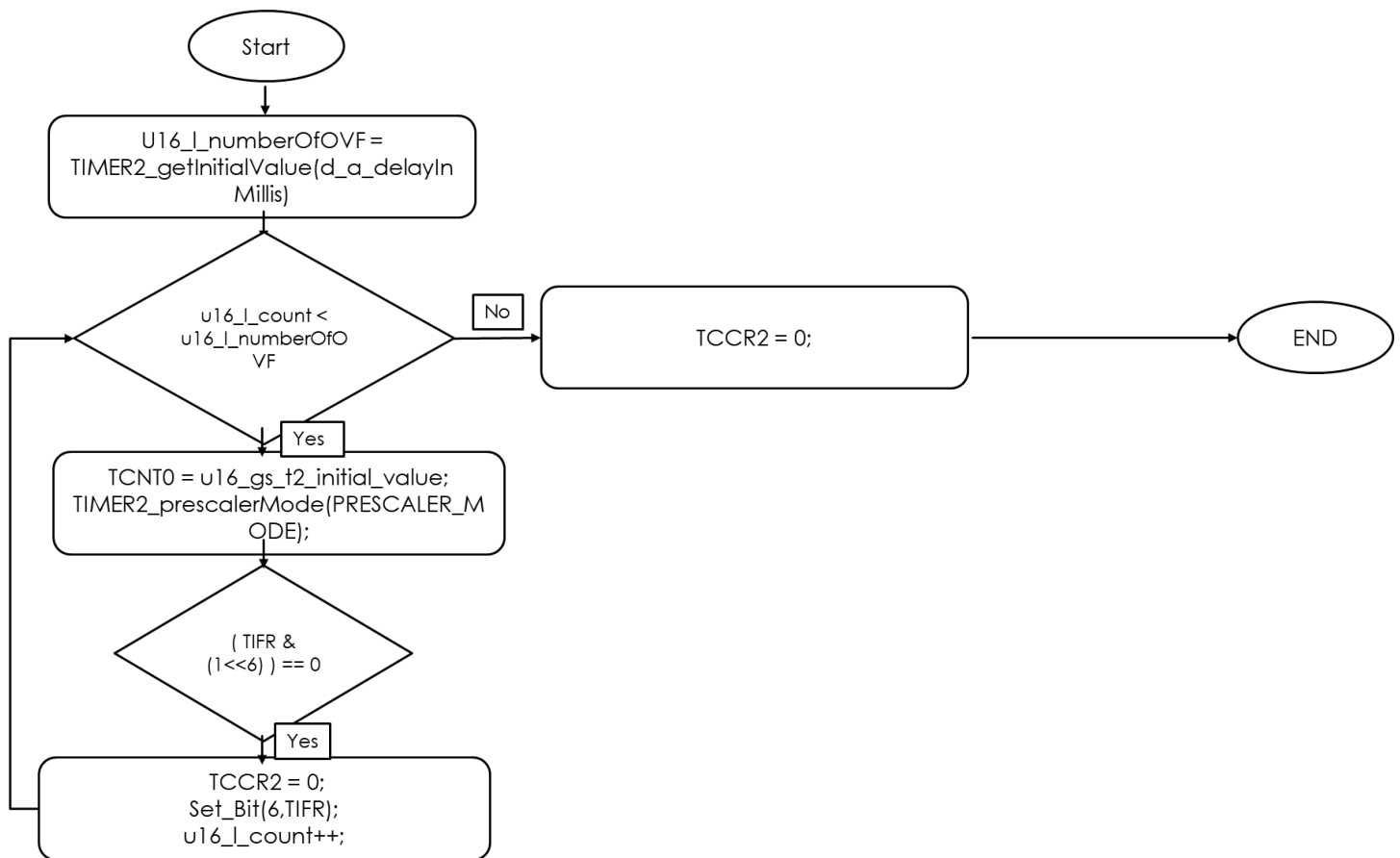




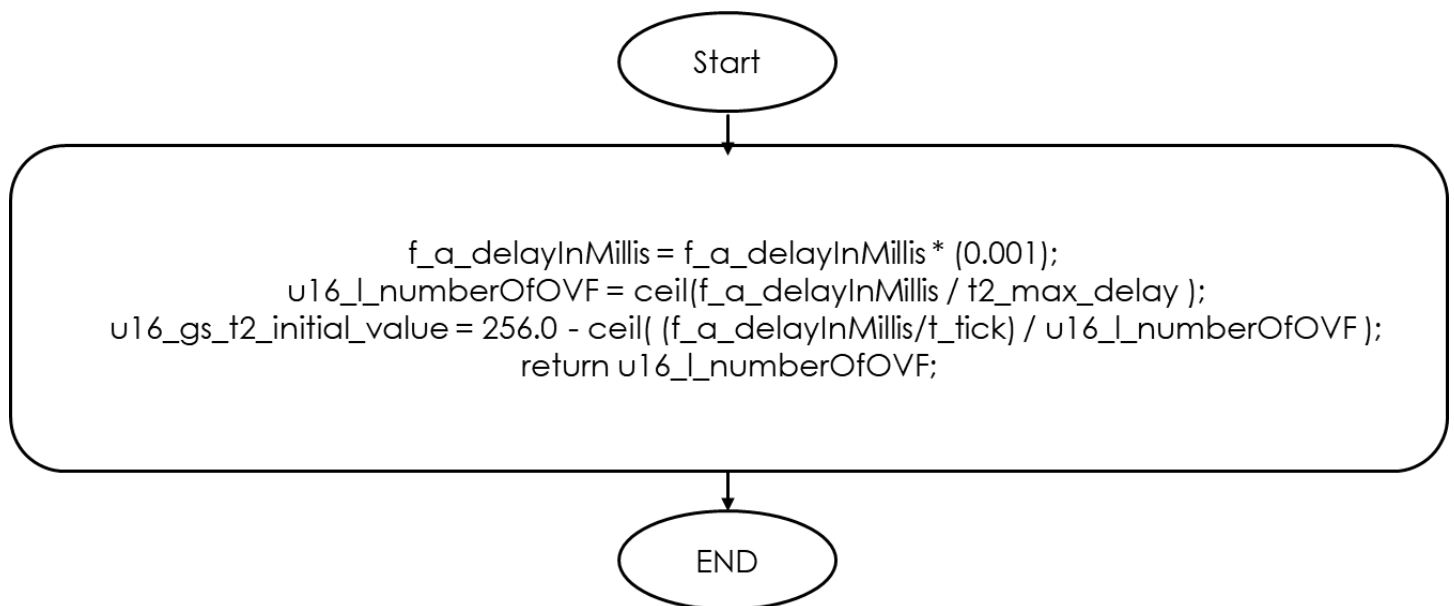
```
err_state TIMER2_prescalerMode(unsigned int u16_a_prescaler);
```



```
err_state TIMER2_delay(float f_a_delayInMillis);
```

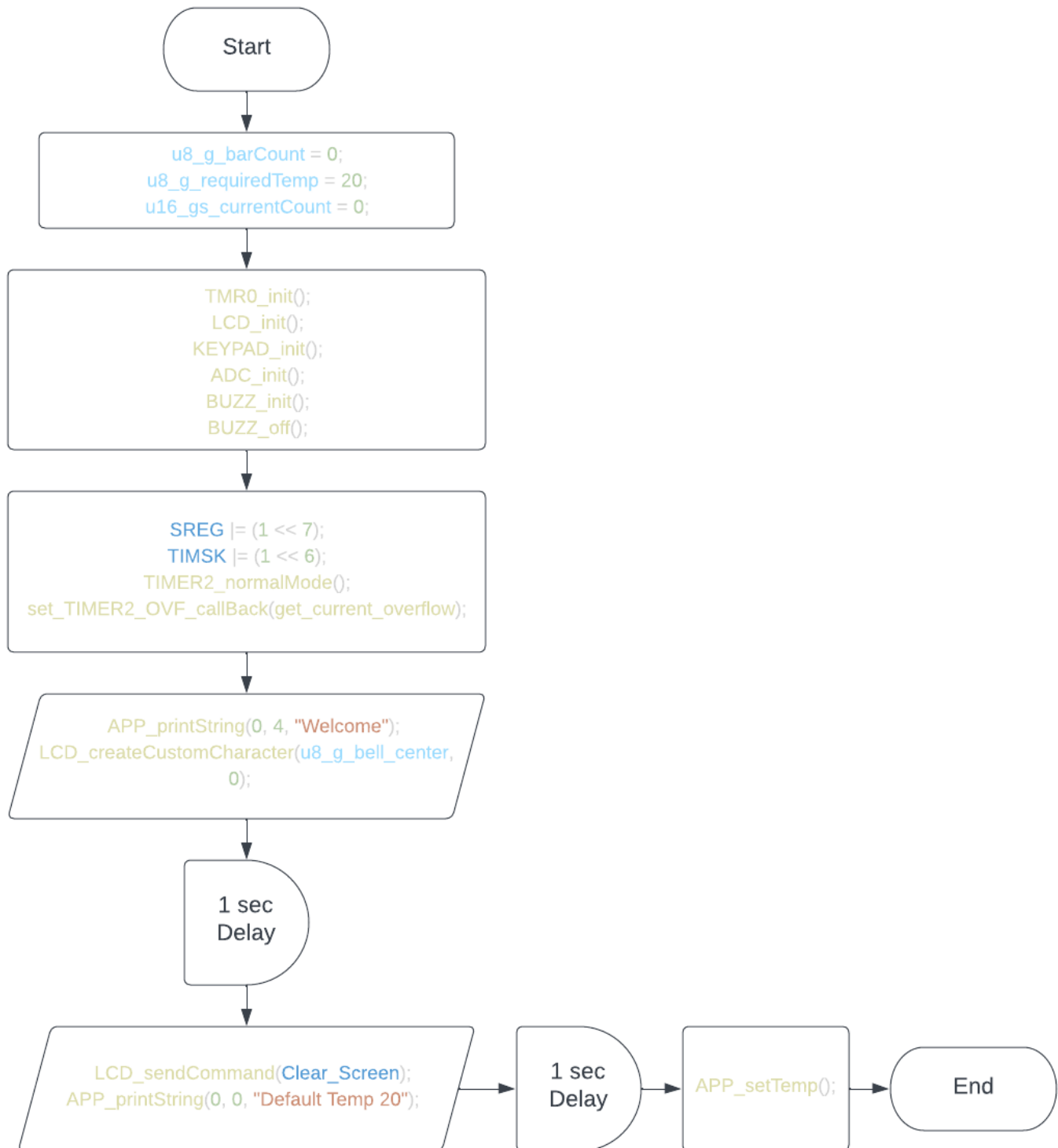


```
unsigned int TIMER2_getInitialValue(float f_a_delayInMillis);
```

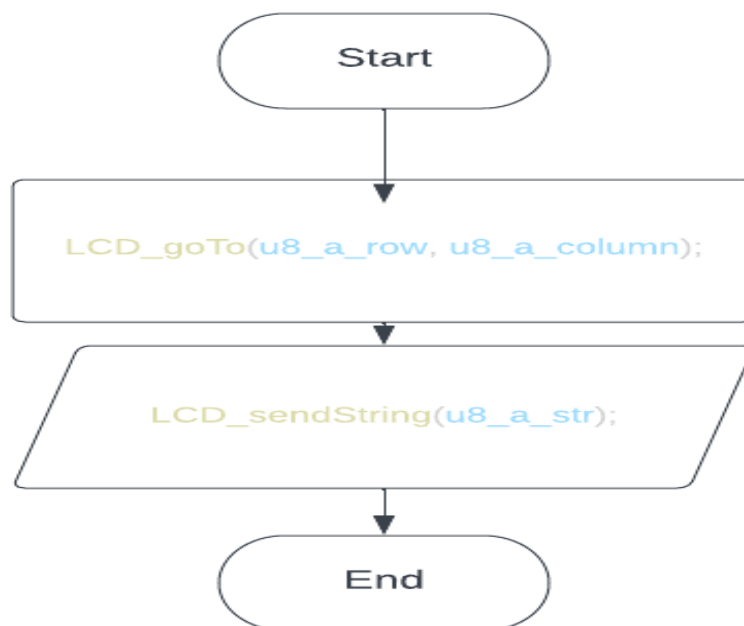


## 7.6 App

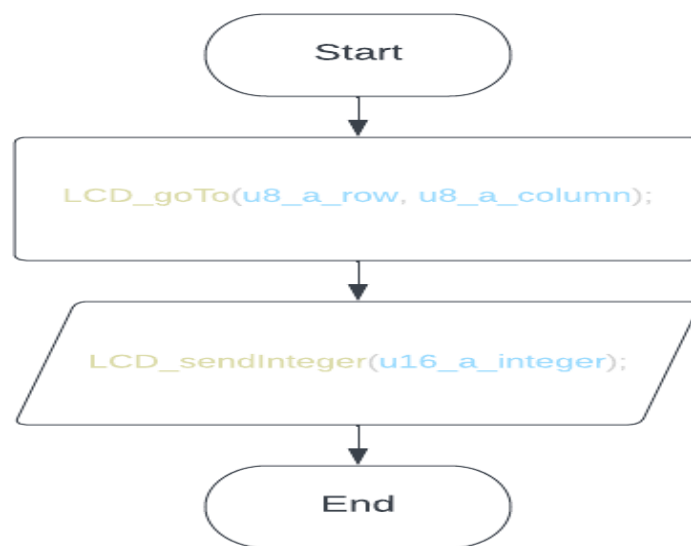
```
initError APP_init(void);
```



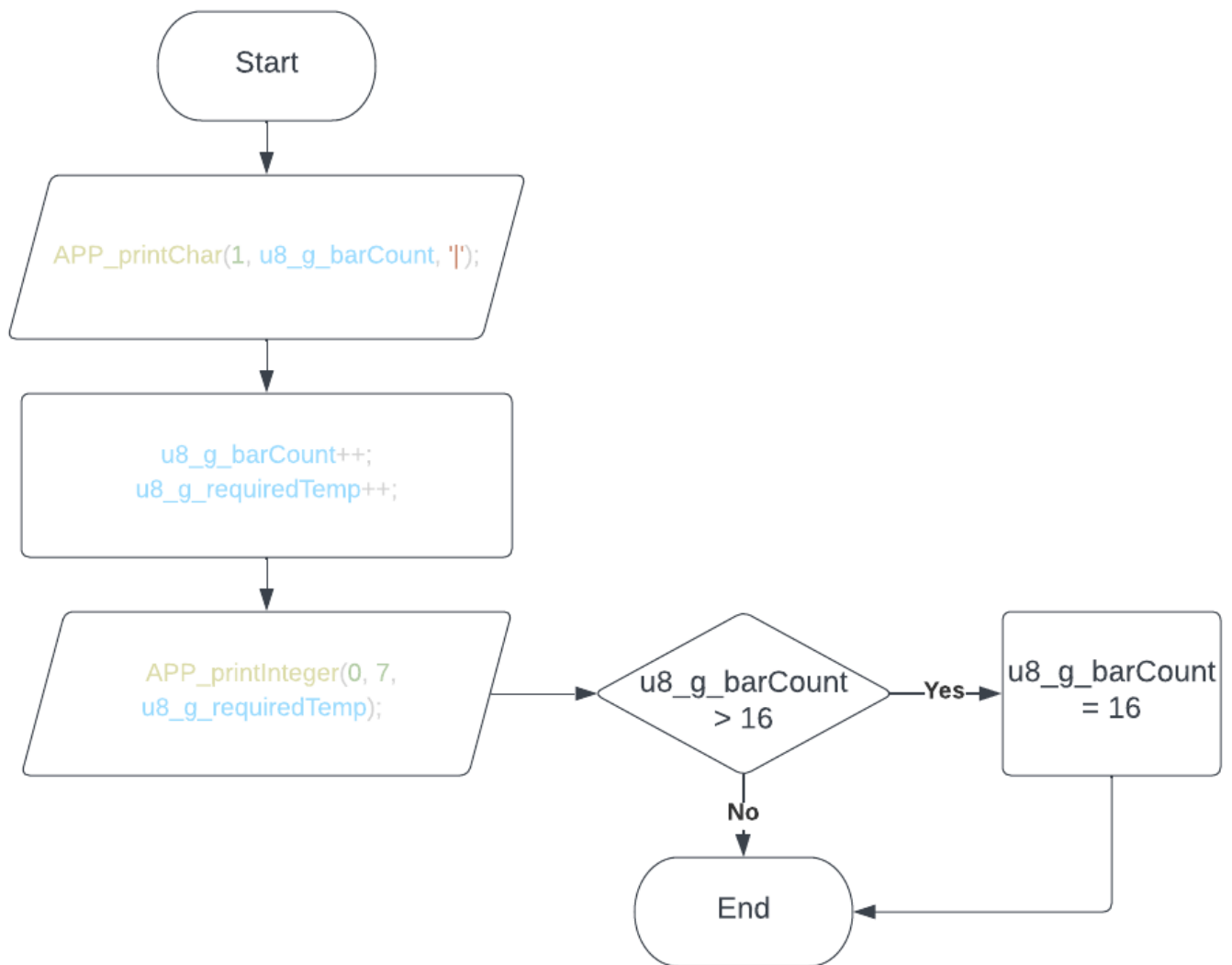
```
void APP_printString(uint8_t u8_a_row, uint8_t u8_a_column, uint8_t *u8_a_str);
```



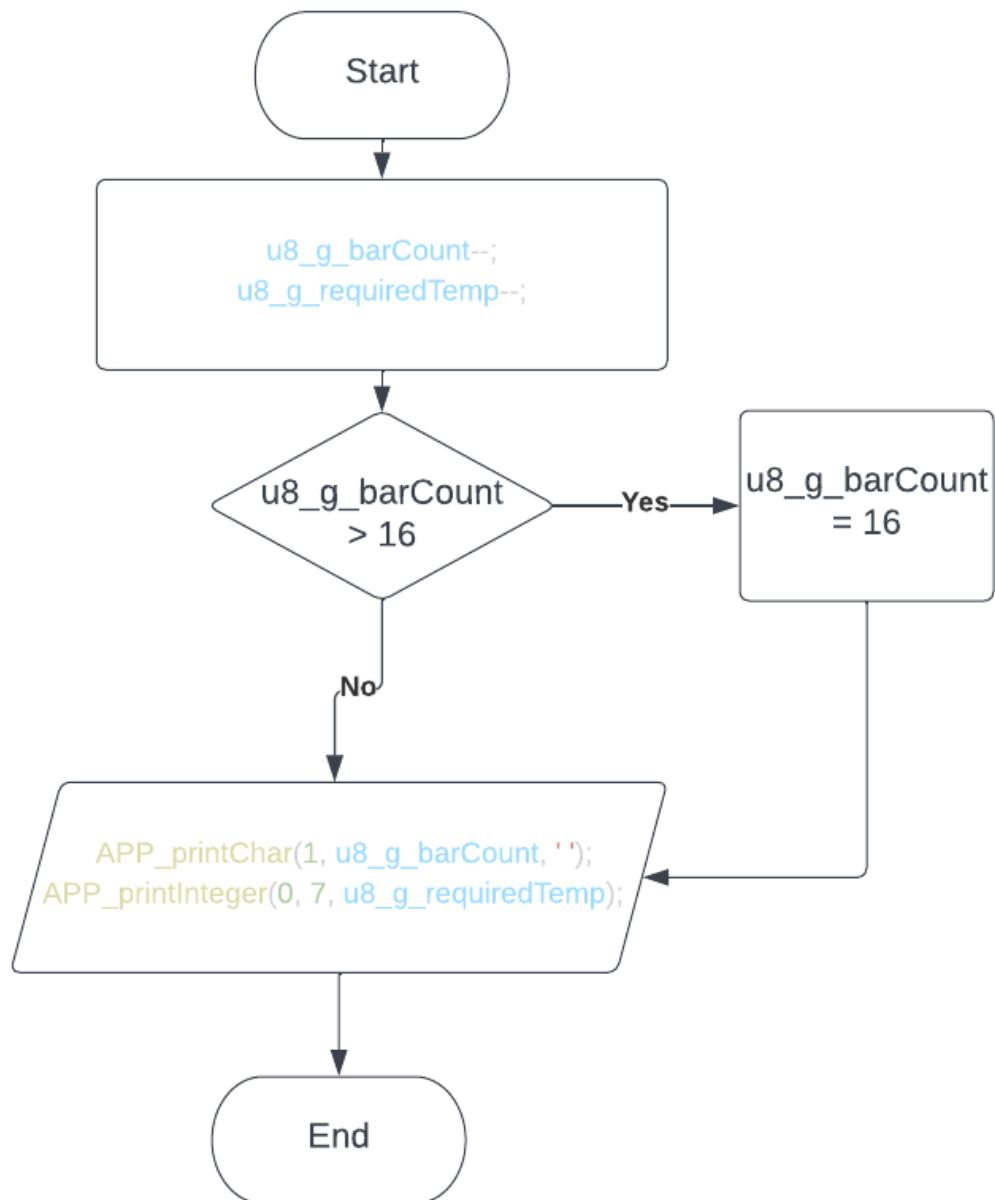
```
printInteger(uint8_t u8_a_row, uint8_t u8_a_column, uint16_t u16_a_integer);
```



```
void APP_incrementBar();
```

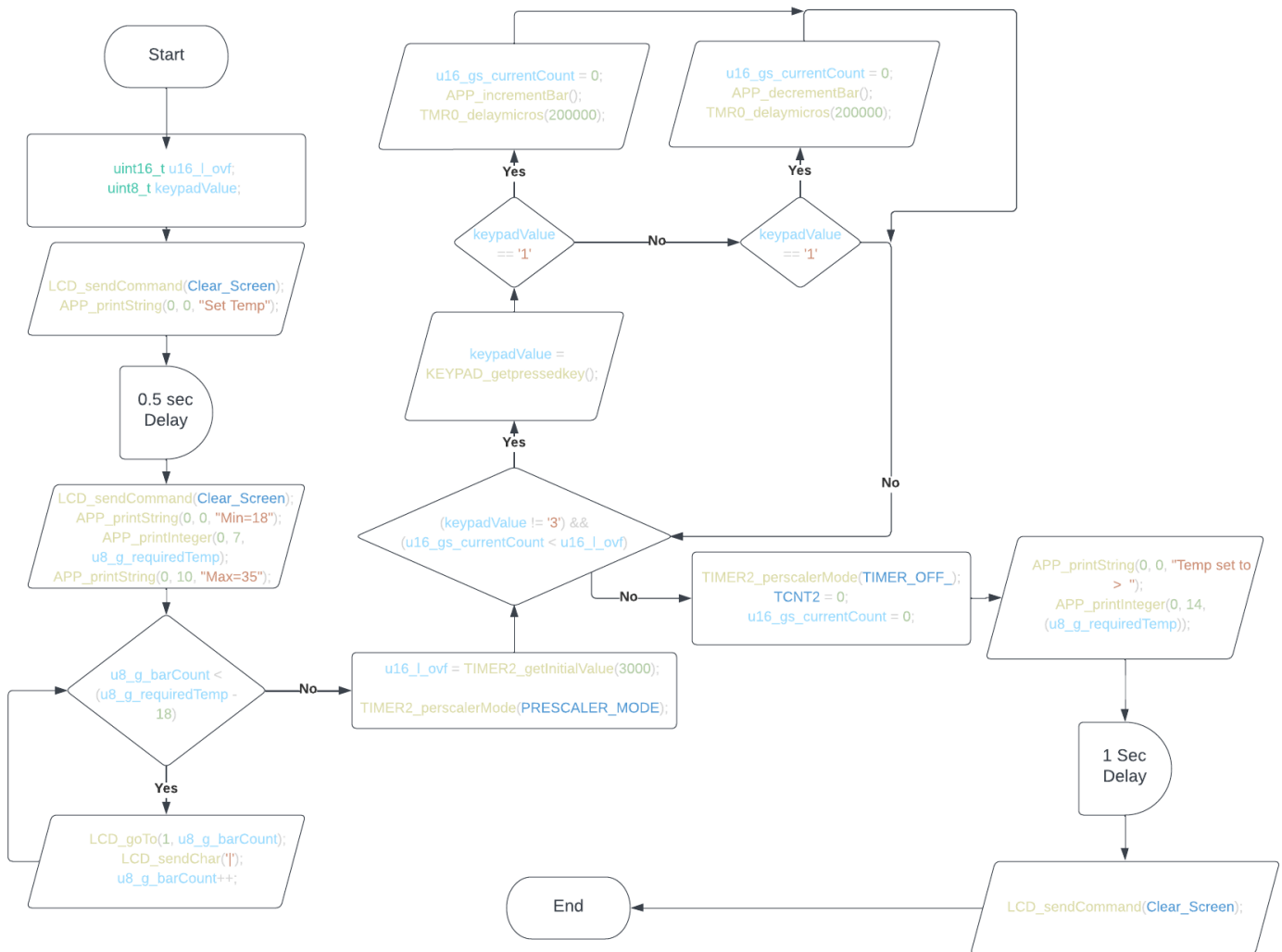


```
void APP_decrementBar();
```

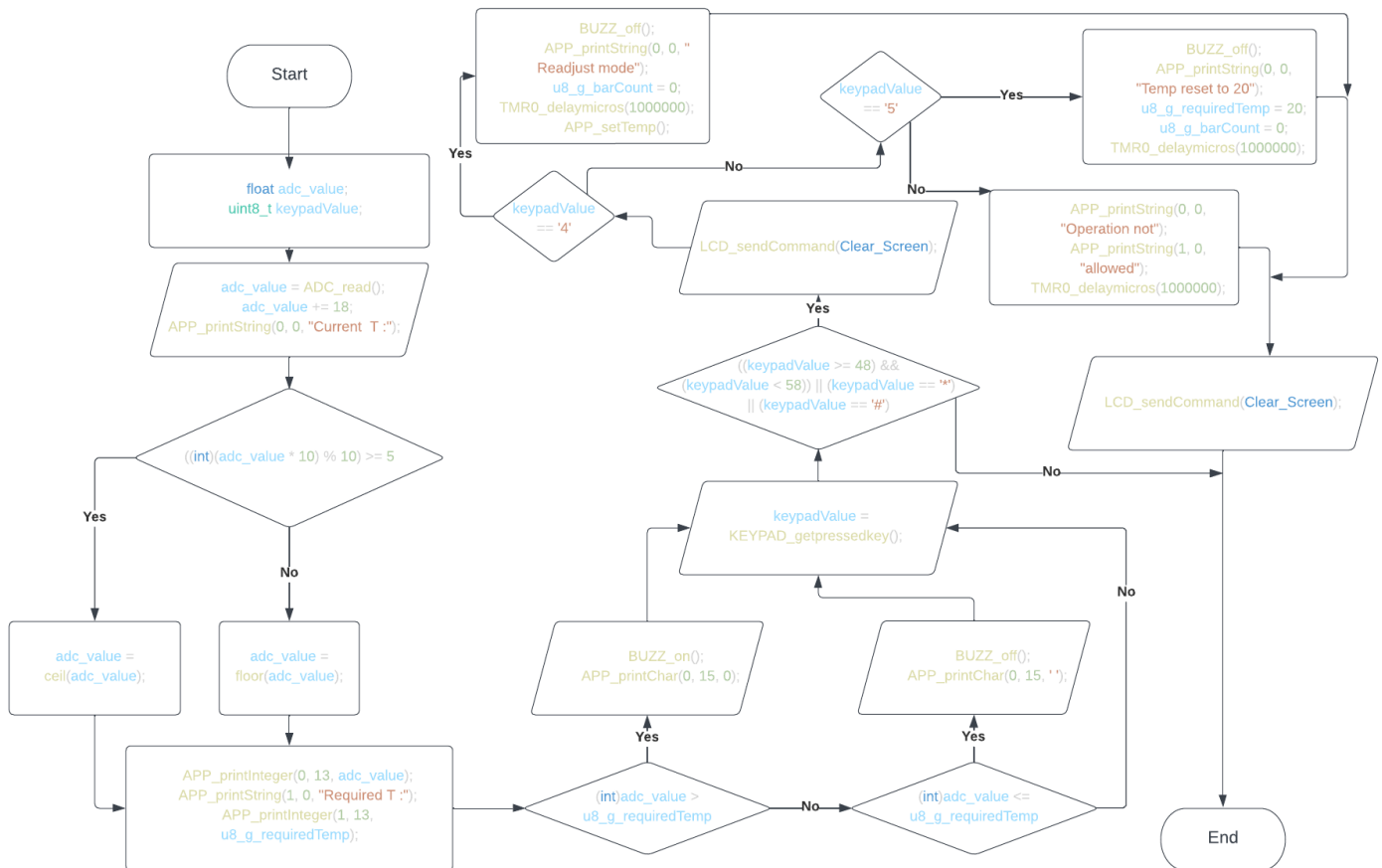




```
void APP_setTemp();
```



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



# Thank You!