**Air Conditioning System**

*Team 2*

*1- Mohab Ahmed*

*2- Anas Mahmoud*

*3- Mustafa Mohammed*

*4- Omar Taha*

Table of Contents

[1-Description 3](#_Toc133505387)

[1.1 Hardware components 3](#_Toc133505388)

[1.2 Software requirements 3](#_Toc133505389)

[2-Layered architecture 4](#_Toc133505390)

[3-System Flow Chart 5](#_Toc133505391)

[4-Schematic Capture 6](#_Toc133505392)

[5-Drivers Description 7](#_Toc133505393)

[5.1 DIO Driver 7](#_Toc133505394)

[5.2 Timer Driver 7](#_Toc133505395)

[5.3 ADC Driver 7](#_Toc133505396)

[5.4 Keypad Driver 7](#_Toc133505397)

[5.5 LCD Driver 8](#_Toc133505398)

[5.6 Application Driver 8](#_Toc133505399)

[6-API’s 9](#_Toc133505400)

[6.1 DIO Driver 9](#_Toc133505401)

[6.2 Timer0 Driver 9](#_Toc133505402)

[6.3 Timer2 Driver 9](#_Toc133505403)

[6.4 ADC Driver 10](#_Toc133505404)

[6.5 Keypad Driver 10](#_Toc133505405)

[6.6 LCD Driver 10](#_Toc133505406)

[6.7 APP Driver 11](#_Toc133505407)

[7- API’s Flow Chart 12](#_Toc133505408)

[7.1 LCD 12](#_Toc133505409)

[7.2 Keypad 25](#_Toc133505410)

[7.3 ADC 27](#_Toc133505411)

[7.4 buzzer 29](#_Toc133505412)

[7.5 Timer 2 31](#_Toc133505413)

[7.6 App 36](#_Toc133505415)

# 1-Description

## 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

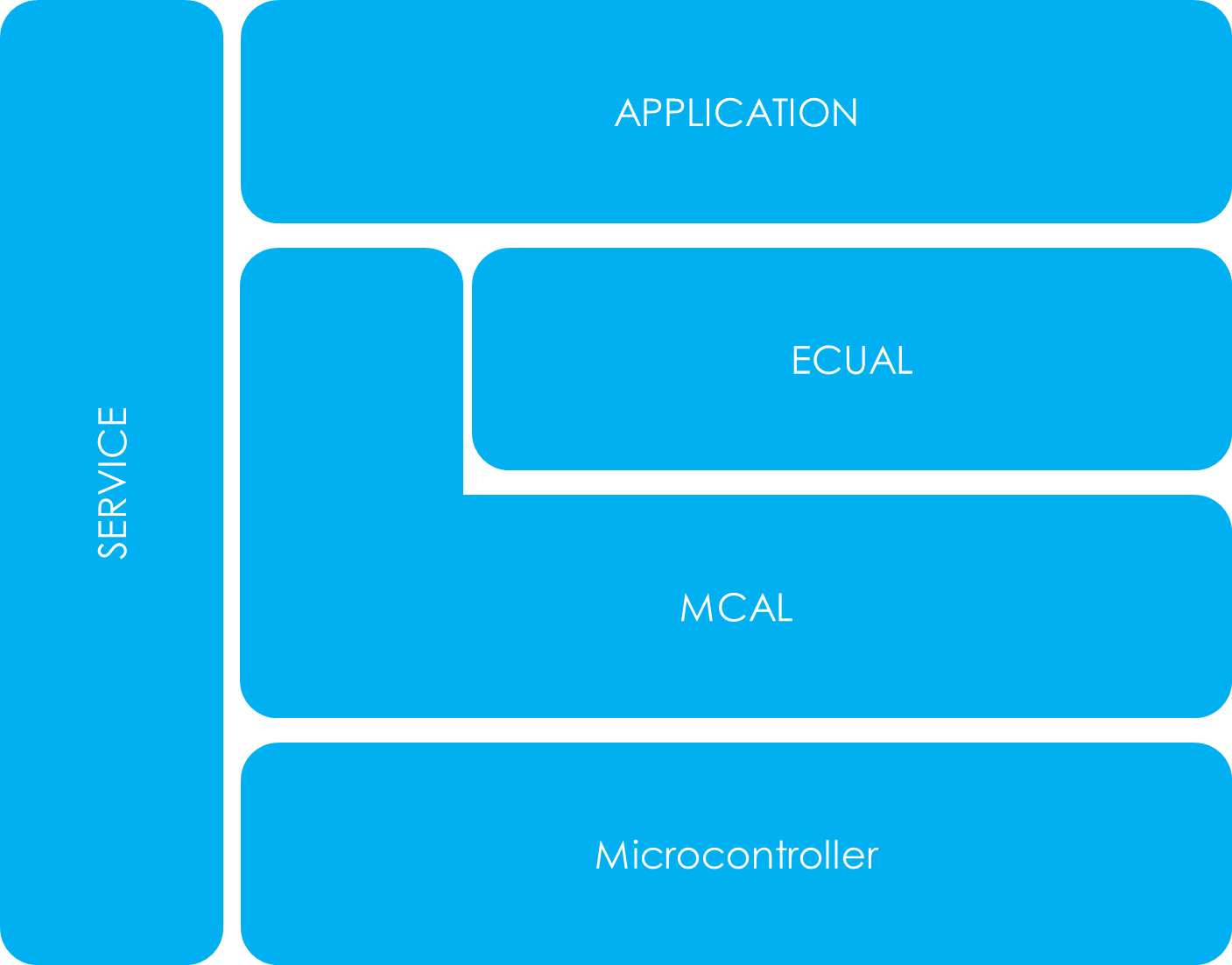
5. Once button three is pressed the desired temperature will be set

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



Registers

Standard\_Types

Common\_macros

Interrupts

Timer0

Timer2

DIO

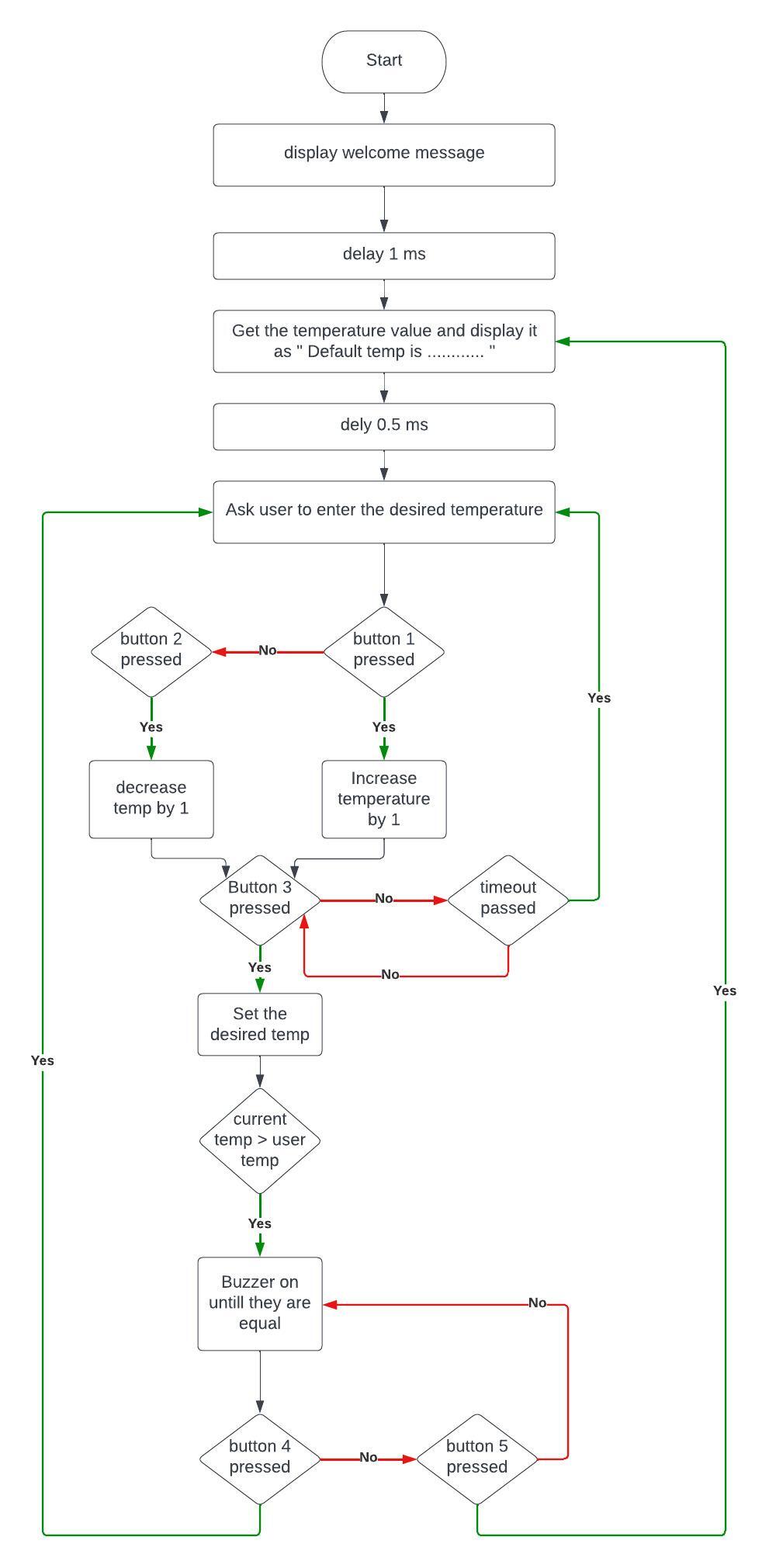
ADC

LCD

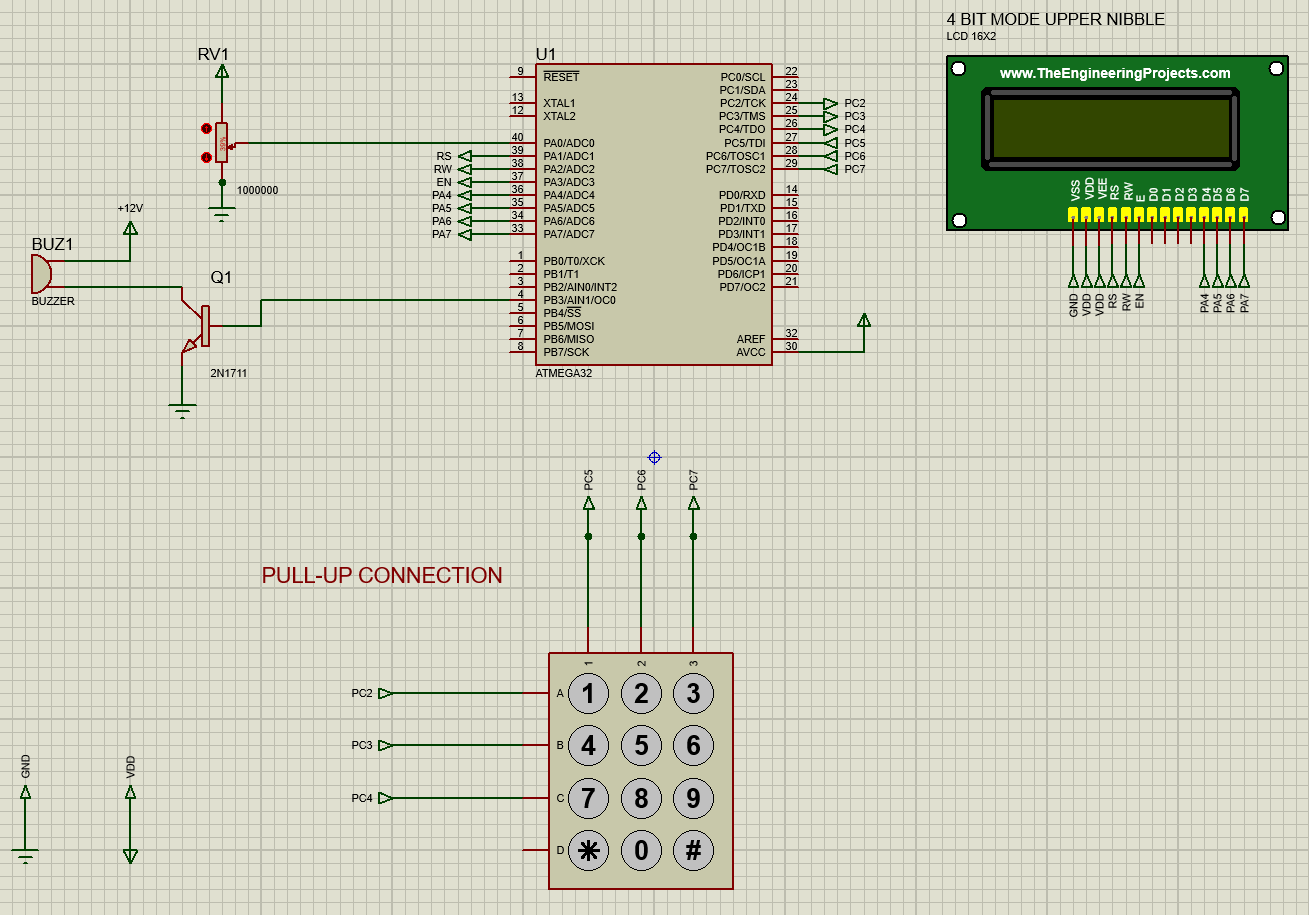
Buzzer

Keypad

# 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);

Diagram

Description automatically generated

LCD\_sendCommand\_error LCD\_8\_bit\_sendCommand(uint8\_t u8\_a\_command);

A screenshot of a computer

Description automatically generated with low confidence

LCD\_sendChar\_error LCD\_8\_bit\_sendChar(uint8\_t u8\_a\_char);

A screenshot of a computer

Description automatically generated with low confidence

LCD\_init\_error LCD\_4\_bit\_init(void);

Diagram, timeline

Description automatically generated with medium confidence

LCD\_sendCommand\_error LCD\_4\_bit\_sendCommand(uint8\_t u8\_a\_command);

A picture containing diagram

Description automatically generated

LCD\_sendChar\_error LCD\_4\_bit\_sendChar(uint8\_t u8\_a\_char);

A picture containing diagram

Description automatically generated

LCD\_sendString\_error LCD\_sendString(uint8\_t \*u8\_a\_string);

Graphical user interface, diagram

Description automatically generated

void LCD\_goTo(uint8\_t u8\_a\_row,uint8\_t u8\_a\_column);

Graphical user interface, application

Description automatically generated

void LCD\_createCustomCharacter(uint8\_t \*u8\_a\_bitMap,uint8\_t u8\_a\_location);

Diagram

Description automatically generated

LCD\_init\_error LCD\_init(void);

Diagram

Description automatically generated

LCD\_sendCommand\_error LCD\_sendCommand(uint8\_t u8\_a\_command);

Graphical user interface, diagram

Description automatically generated

LCD\_sendChar\_error LCD\_sendChar(uint8\_t u8\_a\_char);

Graphical user interface, diagram

Description automatically generated

sendChar\_error LCD\_sendInteger(uint16\_t u16\_a\_number);

Diagram

Description automatically generated

## 7.2 Keypad



Diagram

Description automatically generated



Diagram

Description automatically generated

## 7.3 ADC



Diagram

Description automatically generated

Diagram

Description automatically generated

## 7.4 buzzer



Diagram

Description automatically generatedDiagram

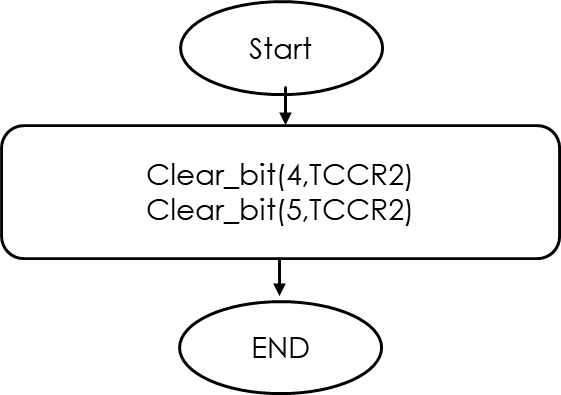
Description automatically generated

Diagram

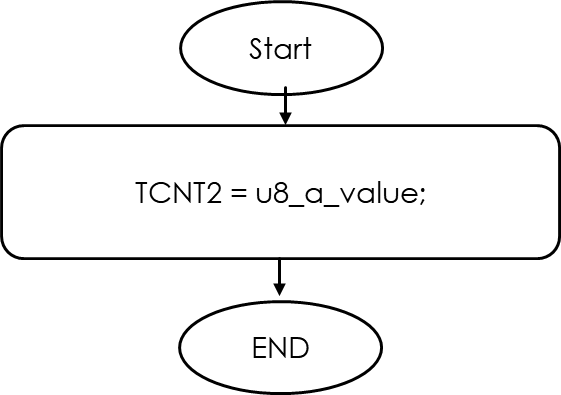
Description automatically generated

## 7.5 Timer 2

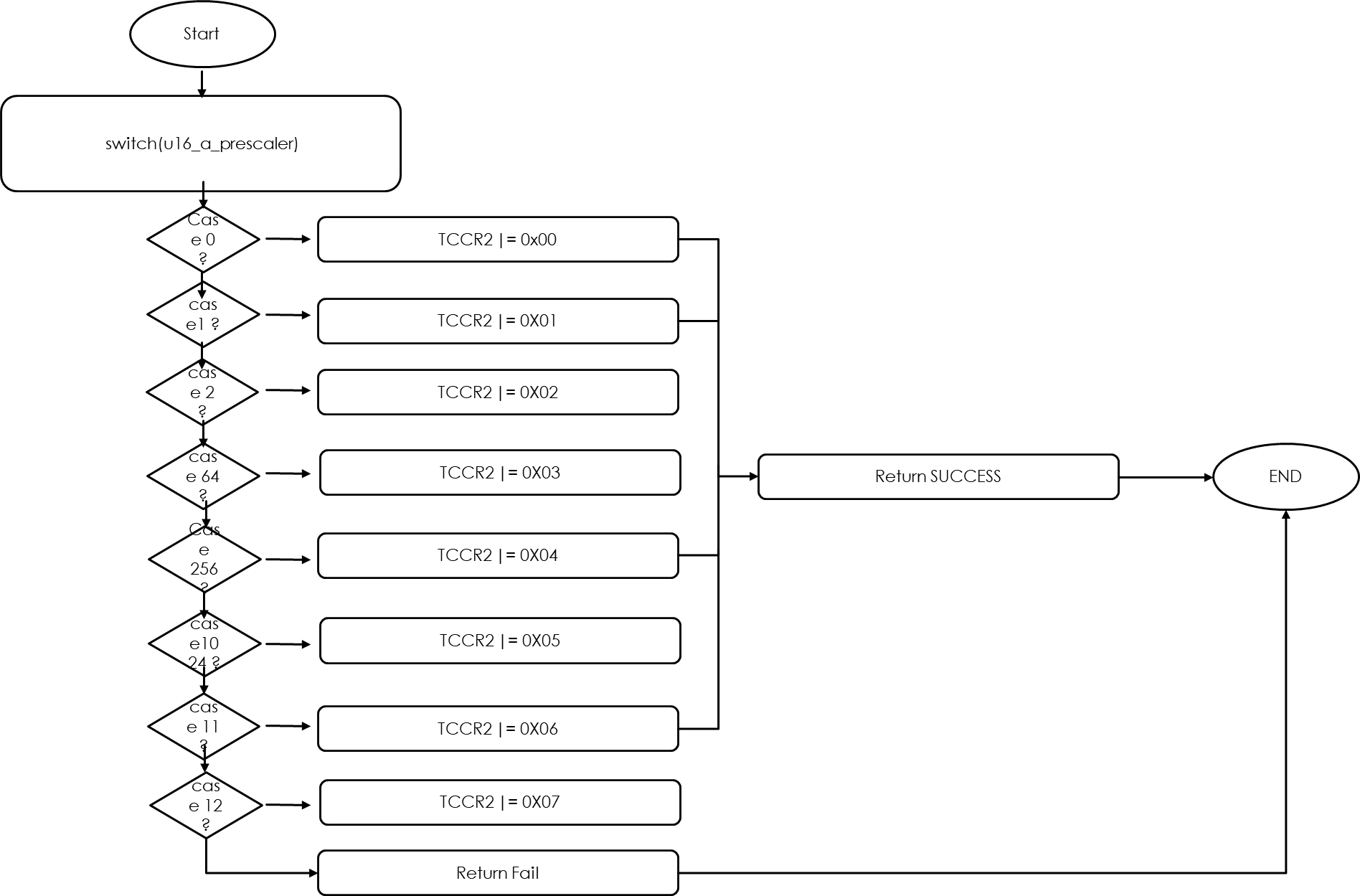
err\_state TIMER2\_normalMode(void);



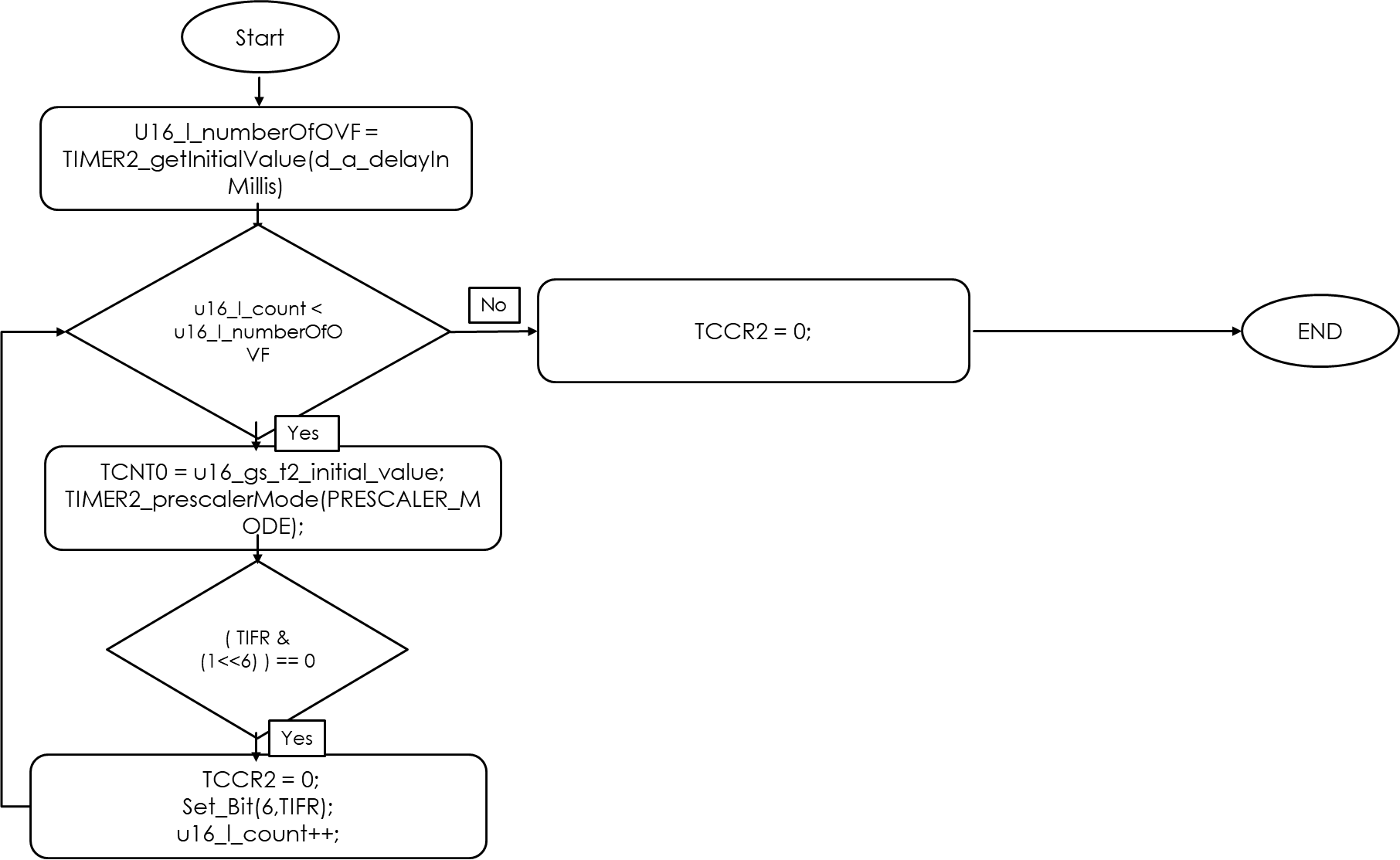
err\_state TIMER2\_initialValue(uint8\_t value);



err\_state TIMER2\_perscalerMode(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);

Graphical user interface

Description automatically generated with medium confidence

void APP\_printString(uint8\_t u8\_a\_row, uint8\_t u8\_a\_column, uint8\_t \*u8\_a\_str);

A screenshot of a computer

Description automatically generated with medium confidence

printInteger(uint8\_t u8\_a\_row, uint8\_t u8\_a\_column, uint16\_t u16\_a\_integer);

A screenshot of a computer

Description automatically generated with medium confidence

void APP\_incrementBar();

Diagram

Description automatically generated

void APP\_decrementBar();

Diagram

Description automatically generated

void APP\_setTemp();

A picture containing diagram

Description automatically generated

void APP\_defaultView(void);

A picture containing diagram

Description automatically generated

Thank You!