Test

1.0.0

Generated by Doxygen 1.8.18

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Data Structure Documentation

3.1 ADC_ConfigType Struct Reference

Data Fields

- ADC_Port_Config **port_config**
  a user defined datatype to select the ADC port configuration
- ADC_Prescalar **prescalar**
  a user defined datatype to select the ADC prescalar

The documentation for this struct was generated from the following file:

- **ADC.h**

3.2 Timer_ConfigType Struct Reference

Data Fields

- Timer **Timer**
  a user defined datatype to select the timer
- Timer0_Prescalar **Timer0_prescalar**
  a user defined datatype to select the timer0 prescalar
- Timer1_Prescalar **Timer1_prescalar**
  a user defined datatype to select the timer1 prescalar
- Timer2_Prescalar **Timer2_prescalar**
  a user defined datatype to select the timer2 prescalar
- Timer0_CLK **Timer0_clk**
  a user defined datatype to select the timer0 clk
- Timer1_CLK **Timer1_clk**
  a user defined datatype to select the timer1 clk
- **uint8 Timer0_value**
  a user defined datatype to select the timer0 initial value
- **uint16 Timer1_value**
  a user defined datatype to select the timer1 initial value
- **uint8 Timer2_value**
  a user defined datatype to select the timer2 initial value

The documentation for this struct was generated from the following file:

- **Timer.h**

Chapter 4

File Documentation

4.1 ADC.c File Reference

ADC Module Source File for this program.

```
#include <pic16f877a.h>
#include "PORT.h"
#include "ADC.h"
```

Functions

- void **ADC_init** (const **ADC_ConfigType** *Config_Ptr)
  Brief: *This is ADC Module Initialization Function*
- **uint16 ADC_readChannel** (**uint8** channel_num)
  Brief: *This is function to read the ADC channel*

4.1.1 Detailed Description

ADC Module Source File for this program.

Author

Nour

Date

12/8/2020

Version

1.0

4.1.2 Function Documentation

4.1.2.1 ADC_init()

```
void ADC_init (
            const  ADC_ConfigType * Config_Ptr )
```

Brief: This is ADC Module Initialization Function

Parameters

Config_Ptr	**ADC_ConfigType** (p. **??**) ∗ Config_Ptr to select Config_Ptr

Returns

void

4.1.2.2 ADC_readChannel()

```
uint16 ADC_readChannel (
            uint8 channel_num )
```

Brief: This is function to read the ADC channel

Parameters

channel_num	uint8 channel_num to select channel_num

Returns

uint16 to get value of ADC channel

References INPUT, PIN0, PIN1, PIN2, PIN3, PIN4, PIN5, PIN6, PIN7, PORTA_CONFIG, PORTE_CONFIG, and set_pin_direction().

Referenced by LM35_sensor_reading().

4.2 ADC.h File Reference

ADC Module Header File for this program.

```
#include "std_types.h"
```

Data Structures

- struct **ADC_ConfigType**

Enumerations

- enum **ADC_Port_Config** {
  _0, **_1**, **_2**, **_3**,
  _4, **_5**, **_6**, **_7**,
  _8, **_9**, **_10**, **_11**,
  _12, **_13**, **_14**, **_15** }
- enum **ADC_Prescalar** {
  Fosc_2, **Fosc_8**, **Fosc_32**, **Frc**,
  Fosc_4, **Fosc_16**, **Fosc_64**, **_Frc** }

Functions

ADC_ConfigType

ADC_ConfigType *(*p. **??***) responsible for dynamic configuration of ADC module*

- void **ADC_init** (const **ADC_ConfigType** ∗Config_Ptr)
  Brief: *This is ADC Module Initialization Function*
- **uint16 ADC_readChannel** (**uint8** channel_num)
  Brief: *This is function to read the ADC channel*

4.2.1 Detailed Description

ADC Module Header File for this program.

Author

Nour

Date

12/8/2020

Version

1.0

4.2.2 Function Documentation

4.2.2.1 ADC_init()

```
void ADC_init (
            const  ADC_ConfigType * Config_Ptr )
```

Brief: This is ADC Module Initialization Function

Parameters

Config_Ptr	**ADC_ConfigType** (p. **??**) ∗ Config_Ptr to select Config_Ptr

Returns

void

4.2.2.2 ADC_readChannel()

```
uint16 ADC_readChannel (
            uint8 channel_num )
```

Brief: This is function to read the ADC channel

Parameters

channel_num	uint8 channel_num to select channel_num

Returns

uint16 to get value of ADC channel

References INPUT, PIN0, PIN1, PIN2, PIN3, PIN4, PIN5, PIN6, PIN7, PORTA_CONFIG, PORTE_CONFIG, and set_pin_direction().

Referenced by LM35_sensor_reading().

4.3 App.c File Reference

App Module Source File for this program.

```
#include "App.h"
#include "seven_segments.h"
#include "LM35_temp_sensor.h"
#include "LEDs.h"
#include "Buttons.h"
#include "EEPROM.h"
#include "Timer.h"
#include "interrupts.h"
#include <xc.h>
```

Macros

- #define **_XTAL_FREQ** 4000000

Functions

- void **off_state** (void)

  Brief: *This is The off state Function*
- void **on_state** (void)

  Brief: *This is The on state Function*
- void **initialize_all** (void)

  Brief: *This is The App Module Initialization Function*

4.3.1 Detailed Description

App Module Source File for this program.

Author

Nour

Date

27/8/2020

Version

1.0

4.3.2 Function Documentation

4.3.2.1 off_state()

```
void off_state (
            void  )
```

Brief: This is The off state Function

Parameters

void	

Returns

void

References display_mode, exit_mode, and read_mode.

4.3.2.2 on_state()

```
void on_state (
            void  )
```

Brief: This is The on state Function

Parameters

void	

Returns

void

References read_button().

4.3.2.3 initialize_all()

```
void initialize_all (
            void  )
```

Brief: This is The App Module Initialization Function

Parameters

void	

Returns

void

References Buttons_init(), EEPROM_Init(), LEDs_init(), LM35_sensor_init(), and seven_segment_init().

4.4 App.h File Reference

App Module Header File for this program.

Functions

- void **initialize_all** (void)

  Brief: *This is The App Module Initialization Function*
- void **on_state** (void)

  Brief: *This is The on state Function*
- void **off_state** (void)

  Brief: *This is The off state Function*

4.4.1 Detailed Description

App Module Header File for this program.

Author

Nour

Date

27/8/2020

Version

1.0

4.4.2 Function Documentation

4.4.2.1 initialize_all()

```
void initialize_all (
            void  )
```

Brief: This is The App Module Initialization Function

Parameters

void	

Returns

void

References Buttons_init(), EEPROM_Init(), LEDs_init(), LM35_sensor_init(), and seven_segment_init().

4.4.2.2 on_state()

```
void on_state (
            void  )
```

Brief: This is The on state Function

Parameters

void	

Returns

void

References read_button().

4.4.2.3 off_state()

```
void off_state (
            void  )
```

Brief: This is The off state Function

Parameters

void	

Returns

void

References display_mode, exit_mode, and read_mode.

4.5 Buttons.c File Reference

Buttons Module Source File for this program.

```
#include "PORT.h"
#include "DIO.h"
#include "Buttons.h"
```

Macros

- #define **UP_BUTTON_PORT PORTD_CONFIG**
  a preprocessor to define up button port
- #define **DOWN_BUTTON_PORT PORTD_CONFIG**
  a preprocessor to define down button port
- #define **ON_OFF_BUTTON_PORT PORTD_CONFIG**
  a preprocessor to define on/off button port
- #define **UP_BUTTON_PIN PIN0**
  a preprocessor to define up button pin
- #define **DOWN_BUTTON_PIN PIN1**
  a preprocessor to define down button pin
- #define **ON_OFF_BUTTON_PIN PIN2**
  a preprocessor to define on/off button pin

Functions

- void **Buttons_init** (void)

  Brief: *This is The Buttons Module Initialization Function*
- **uint8 read_button** (Buttons button)

  Brief: *This is function to read the button*

4.5.1 Detailed Description

Buttons Module Source File for this program.

Author

Nour

Date

16/8/2020

Version

1.0

4.5.2 Function Documentation

4.5.2.1 Buttons_init()

```
void Buttons_init (
             void  )
```

Brief: This is The Buttons Module Initialization Function

Parameters

void	

Returns

void

References DOWN_BUTTON_PIN, DOWN_BUTTON_PORT, INPUT, ON_OFF_BUTTON_PIN, ON_OFF_BUTT↩ ON_PORT, set_pin_direction(), UP_BUTTON_PIN, and UP_BUTTON_PORT.

Referenced by initialize_all().

4.5.2.2 read_button()

```
uint8 read_button (
            Buttons button )
```

Brief: This is function to read the button

Parameters

button	Buttons button to select button

Returns

uint8 to get value of button

Referenced by on_state().

4.6 Buttons.h File Reference

Buttons Module Header File for this program.

```
#include "std_types.h"
```

Enumerations

- enum **Buttons** { **Up_Button**, **Down_Button**, **On_Off_Button** }

Functions

- void **Buttons_init** (void)
  Brief: *This is The Buttons Module Initialization Function*
- **uint8 read_button** (Buttons button)
  Brief: *This is function to read the button*

4.6.1 Detailed Description

Buttons Module Header File for this program.

Author

Nour

Date

16/8/2020

Version

1.0

4.6.2 Function Documentation

4.6.2.1 Buttons_init()

```
void Buttons_init (
            void  )
```

Brief: This is The Buttons Module Initialization Function

Parameters

void	

Returns

void

References DOWN_BUTTON_PIN, DOWN_BUTTON_PORT, INPUT, ON_OFF_BUTTON_PIN, ON_OFF_BUTT↩ ON_PORT, set_pin_direction(), UP_BUTTON_PIN, and UP_BUTTON_PORT.

Referenced by initialize_all().

4.6.2.2 read_button()

```
uint8 read_button (
            Buttons button )
```

Brief: This is function to read the button

Parameters

button	Buttons button to select button

Returns

uint8 to get value of button

Referenced by on_state().

4.7 DIO.c File Reference

DIO Module Source File for this program.

```
#include "DIO.h"
#include <pic16f877a.h>
```

Functions

- void **write_pin** (**uint8** port, **uint8** pin, **uint8** value)

  Brief: *This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to set the value (LOW , HIGH)*
- void **toggle_pin** (**uint8** port, **uint8** pin)

  Brief: *This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to toggle it*
- **uint8 read_pin** (**uint8** port, **uint8** pin)

  Brief: *This is function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to get the value*
- void **write_group_value** (**uint8** port, **uint8** group, **uint8** value)

  Brief: *This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set a certain value*
- void **write_port** (**uint8** port, **uint8** value)

  Brief: *This is a function to select certain port of ports (A,B,C,D,E) to set the value (LOW , HIGH , any value)*
- void **toggle_port** (**uint8** port)

  Brief: *This is a function to select certain port of ports (A,B,C,D,E) to toggle it*
- void **write_group** (**uint8** port, **uint8** group, **uint8** value)

  Brief: *This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set the value (LOW , HIGH)*
- void **toggle_group** (**uint8** port, **uint8** group)

  Brief: *This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to toggle it*

4.7.1 Detailed Description

DIO Module Source File for this program.

Author

Nour

Date

27/7/2020

Version

1.0

4.7.2 Function Documentation

4.7.2.1 write_pin()

```
void write_pin (
              uint8 port,
              uint8 pin,
              uint8 value )
```

Brief: This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to set the value (LOW , HIGH)

Parameters

port	uint8 port to select port
pin	uint8 pin to select pin
value	uint8 value to select value

Returns

void

References HIGH, LOW, PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by display_num_2_7_seg(), LEDs_init(), seven_segment_init(), and turn_off_2_7_seg().

4.7.2.2 toggle_pin()

```
void toggle_pin (
              uint8 port,
              uint8 pin )
```

Brief: This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to toggle it

Parameters

port	uint8 port to select port
pin	uint8 pin to select pin

Returns

void

References PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by blink_LED().

4.7.2.3 read_pin()

```
uint8 read_pin (
              uint8 port,
              uint8 pin )
```

Brief: This is function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to get the value

Parameters

port	uint8 port to select port
pin	uint8 pin to select pin

Returns

uint8 to get value of pin

References PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

4.7.2.4 write_group_value()

```
void write_group_value (
              uint8 port,
              uint8 group,
              uint8 value )
```

Brief: This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set a certain value

Parameters

port	uint8 port to select port
group	uint8 group to select group
value	uint8 value to select value

Returns

void

References PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by display_num().

4.7.2.5 write_port()

```
void write_port (
              uint8 port,
              uint8 value )
```

Brief: This is a function to select certain port of ports (A,B,C,D,E) to set the value (LOW , HIGH , any value)

Parameters

port	uint8 port to select port
value	uint8 value to select value

Returns

void

References HIGH, LOW, PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by display_num(), and seven_segment_init().

4.7.2.6 toggle_port()

```
void toggle_port (
              uint8 port )
```

Brief: This is a function to select certain port of ports (A,B,C,D,E) to toggle it

Parameters

port	uint8 port to select port

Returns

void

References PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

4.7.2.7 write_group()

```
void write_group (
              uint8 port,
              uint8 group,
              uint8 value )
```

Brief: This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set the value (LOW , HIGH)

Parameters

port	uint8 port to select port
group	uint8 group to select group
value	uint8 value to select value

Returns

void

References HIGH, LOW, PORTB_CONFIG, PORTC_CONFIG, and PORTD_CONFIG.

Referenced by seven_segment_init().

4.7.2.8 toggle_group()

```
void toggle_group (
              uint8 port,
              uint8 group )
```

Brief: This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to toggle it

Parameters

port	uint8 port to select port
group	uint8 group to select group

Returns

void

References PORTB_CONFIG, PORTC_CONFIG, and PORTD_CONFIG.

4.8 DIO.h File Reference

DIO Module Header File for this program.

```
#include "std_types.h"
#include "PORT_config.h"
#include "PIN_config.h"
```

Functions

- void **write_pin** (**uint8** port, **uint8** pin, **uint8** value)

  Brief: *This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to set the value (LOW , HIGH)*
- void **toggle_pin** (**uint8** port, **uint8** pin)

  Brief: *This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to toggle it*
- **uint8 read_pin** (**uint8** port, **uint8** pin)

  Brief: *This is function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to get the value*
- void **write_group_value** (**uint8** port, **uint8** group, **uint8** value)

Brief: *This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set a certain value*

- void **write_port** (**uint8** port, **uint8** value)

  Brief: *This is a function to select certain port of ports (A,B,C,D,E) to set the value (LOW , HIGH , any value)*

- void **toggle_port** (**uint8** port)

  Brief: *This is a function to select certain port of ports (A,B,C,D,E) to toggle it*

- void **write_group** (**uint8** port, **uint8** group, **uint8** value)

  Brief: *This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set the value (LOW , HIGH)*

- void **toggle_group** (**uint8** port, **uint8** group)

  Brief: *This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to toggle it*

4.8.1 Detailed Description

DIO Module Header File for this program.

Author

Nour

Date

27/7/2020

Version

1.0

4.8.2 Function Documentation

4.8.2.1 write_pin()

```
void write_pin (
              uint8 port,
              uint8 pin,
              uint8 value )
```

Brief: This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to set the value (LOW , HIGH)

Parameters

port	uint8 port to select port
pin	uint8 pin to select pin
value	uint8 value to select value

Returns

void

References HIGH, LOW, PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by display_num_2_7_seg(), LEDs_init(), seven_segment_init(), and turn_off_2_7_seg().

4.8.2.2 toggle_pin()

```
void toggle_pin (
              uint8 port,
              uint8 pin )
```

Brief: This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to toggle it

Parameters

port	uint8 port to select port
pin	uint8 pin to select pin

Returns

void

References PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by blink_LED().

4.8.2.3 read_pin()

```
uint8 read_pin (
              uint8 port,
              uint8 pin )
```

Brief: This is function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to get the value

Parameters

port	uint8 port to select port
pin	uint8 pin to select pin

Returns

uint8 to get value of pin

References PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

4.8.2.4 write_group_value()

```
void write_group_value (
              uint8 port,
              uint8 group,
              uint8 value )
```

Brief: This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set a certain value

Parameters

port	uint8 port to select port
group	uint8 group to select group
value	uint8 value to select value

Returns

void

References PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by display_num().

4.8.2.5 write_port()

```
void write_port (
              uint8 port,
              uint8 value )
```

Brief: This is a function to select certain port of ports (A,B,C,D,E) to set the value (LOW , HIGH , any value)

Parameters

port	uint8 port to select port
value	uint8 value to select value

Returns

void

References HIGH, LOW, PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORT↩E_CONFIG.

Referenced by display_num(), and seven_segment_init().

4.8.2.6 toggle_port()

```
void toggle_port (
              uint8 port )
```

Brief: This is a function to select certain port of ports (A,B,C,D,E) to toggle it

Parameters

port	uint8 port to select port

Returns

void

References PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

4.8.2.7 write_group()

```
void write_group (
              uint8 port,
              uint8 group,
              uint8 value )
```

Brief: This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set the value (LOW , HIGH)

Parameters

port	uint8 port to select port
group	uint8 group to select group
value	uint8 value to select value

Returns

void

References HIGH, LOW, PORTB_CONFIG, PORTC_CONFIG, and PORTD_CONFIG.

Referenced by seven_segment_init().

4.8.2.8 toggle_group()

```
void toggle_group (
              uint8 port,
              uint8 group )
```

Brief: This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to toggle it

Parameters

port	uint8 port to select port
group	uint8 group to select group

Returns

void

References PORTB_CONFIG, PORTC_CONFIG, and PORTD_CONFIG.

4.9 EEPROM.c File Reference

EEPROM Module Source File for this program.

```
#include "EEPROM.h"
#include "I2C.h"
```

Macros

- #define **EEPROM_ADDRESS** 0x50
  EEPROM Address.

Functions

- void **EEPROM_Init** (void)
  Brief: *This is the Initialization of EEPROM function to Intialize ECU as Maseter*
- void **EEPROM_Write** (**uint8** address, **uint8** data)
  Brief: *This is the External EEPROM Write function to write a certain data at certain address of external EEPROM*
- **uint8 EEPROM_Read** (**uint8** address)
  Brief: *This is the External EEPROM Read function to read a certain data at certain address of external EEPROM*

4.9.1 Detailed Description

EEPROM Module Source File for this program.

Author

Nour

Date

10/8/2020

Version

1.0

4.9.2 Function Documentation

4.9.2.1 EEPROM_Init()

```
void EEPROM_Init (
            void  )
```

Brief: This is the Initialization of EEPROM function to Intialize ECU as Maseter

Parameters

void	

Returns

void

References I2C_Master_Init().

Referenced by initialize_all().

4.9.2.2 EEPROM_Write()

```
void EEPROM_Write (
              uint8 address,
              uint8 data )
```

Brief: This is the External EEPROM Write function to write a certain data at certain address of external EEPROM

Parameters

address	uint8 address to select a certain address
data	uint8 data to set data

Returns

void

References EEPROM_ADDRESS, I2C_Master_write_byte(), I2C_Master_write_slave_address_with_write_req(), I2C_Restart(), I2C_Start(), and I2C_Stop().

4.9.2.3 EEPROM_Read()

```
uint8 EEPROM_Read (
            uint8 address )
```

Brief: This is the External EEPROM Read function to read a certain data at certain address of external EEPROM

Parameters

address	uint8 address to select a certain address

Returns

uint8 to read data

References EEPROM_ADDRESS, I2C_Master_read_byte(), I2C_Master_write_byte(), I2C_Master_write_slave_address_with_read_req(), I2C_Master_write_slave_address_with_write_req(), I2C_NAck(), I2C_Restart(), I2C_Start(), and I2C_Stop().

4.10 EEPROM.h File Reference

EEPROM Module header file for this program.

```
#include "std_types.h"
```

Functions

- void **EEPROM_Init** (void)
  Brief: *This is the Initialization of EEPROM function to Intialize ECU as Maseter*
- void **EEPROM_Write** (**uint8** address, **uint8** data)
  Brief: *This is the External EEPROM Write function to write a certain data at certain address of external EEPROM*
- **uint8 EEPROM_Read** (**uint8** address)
  Brief: *This is the External EEPROM Read function to read a certain data at certain address of external EEPROM*

4.10.1 Detailed Description

EEPROM Module header file for this program.

Author

Nour

Date

10/8/2020

Version

1.0

4.10.2 Function Documentation

4.10.2.1 EEPROM_Init()

```
void EEPROM_Init (
            void  )
```

Brief: This is the Initialization of EEPROM function to Intialize ECU as Maseter

Parameters

void	

Returns

void

References I2C_Master_Init().

Referenced by initialize_all().

4.10.2.2 EEPROM_Write()

```
void EEPROM_Write (
              uint8 address,
              uint8 data )
```

Brief: This is the External EEPROM Write function to write a certain data at certain address of external EEPROM

Parameters

address	uint8 address to select a certain address
data	uint8 data to set data

Returns

void

References EEPROM_ADDRESS, I2C_Master_write_byte(), I2C_Master_write_slave_address_with_write_req(), I2C_Restart(), I2C_Start(), and I2C_Stop().

4.10.2.3 EEPROM_Read()

```
uint8 EEPROM_Read (
            uint8 address )
```

Brief: This is the External EEPROM Read function to read a certain data at certain address of external EEPROM

Parameters

address	uint8 address to select a certain address

Returns

uint8 to read data

References EEPROM_ADDRESS, I2C_Master_read_byte(), I2C_Master_write_byte(), I2C_Master_write_slave↩ _address_with_read_req(), I2C_Master_write_slave_address_with_write_req(), I2C_NAck(), I2C_Restart(), I2C_↩ Start(), and I2C_Stop().

4.11 I2C.c File Reference

I2C Module Source File for this program.

```
#include "I2C.h"
#include <pic16f877a.h>
```

Macros

- #define **SCL_PIN** 3
  I2C Clock Pin.
- #define **SDA_PIN** 4
  I2C Data Pin.
- #define **_XTAL_FREQ** 4000000
  Clock Frequency.
- #define **I2C_BAUDRATE** 9600
  I2C Baud Rate.

Functions

- void **I2C_Master_Init** (void)
  Brief: *This is the function to initialize ECU as Master Mode*
- void **I2C_Start** (void)
  Brief: *This is the function to Start I2C communication protocol*
- void **I2C_Stop** (void)
  Brief: *This is the function to Stop I2C communication protocol*
- void **I2C_Restart** (void)

Brief: *This is the function to Restart I2C communication protocol*

- void **I2C_Wait** (void)

  Brief: *This is the I2C wait function*

- void **I2C_NAck** (void)

  Brief: *This is the I2C not Ack function*

- **uint8 I2C_Master_write_slave_address_with_write_req** (**uint8** address)

  Brief: *This is function to Master write address byte with write request*

- **uint8 I2C_Master_write_slave_address_with_read_req** (**uint8** address)

  Brief: *This is function to Master write address byte with read request*

- **uint8 I2C_Master_write_byte** (**uint8** data)

  Brief: *This is function to Master write data byte*

- **uint8 I2C_Master_read_byte** (void)

  Brief: *This is function to Master read data byte*

4.11.1 Detailed Description

I2C Module Source File for this program.

Author

Nour

Date

10/8/2020

Version

1.0

4.11.2 Function Documentation

4.11.2.1 I2C_Master_Init()

```
void I2C_Master_Init (
            void  )
```

Brief: This is the function to initialize ECU as Master Mode

Parameters

void	

Returns

void

References _XTAL_FREQ, and I2C_BAUDRATE.

Referenced by EEPROM_Init().

4.11.2.2 I2C_Start()

```
void I2C_Start (
            void  )
```

Brief: This is the function to Start I2C communication protocol

Parameters

void	

Returns

void

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.11.2.3 I2C_Stop()

```
void I2C_Stop (
            void  )
```

Brief: This is the function to Stop I2C communication protocol

Parameters

void	

Returns

void

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.11.2.4 I2C_Restart()

```
void I2C_Restart (
            void  )
```

Brief: This is the function to Restart I2C communication protocol

Parameters

void	

Returns

void

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.11.2.5 I2C_Wait()

```
void I2C_Wait (
            void  )
```

Brief: This is the I2C wait function

Parameters

void	

Returns

void

Referenced by I2C_Master_read_byte(), I2C_Master_write_byte(), I2C_Master_write_slave_address_with_read↩_req(), I2C_Master_write_slave_address_with_write_req(), I2C_NAck(), I2C_Restart(), I2C_Start(), and I2C_↩Stop().

4.11.2.6 I2C_NAck()

```
void I2C_NAck (
            void  )
```

Brief: This is the I2C not Ack function

Parameters

void	

Returns

void

References I2C_Wait().

Referenced by EEPROM_Read().

4.11.2.7 I2C_Master_write_slave_address_with_write_req()

```
uint8 I2C_Master_write_slave_address_with_write_req (
            uint8 address )
```

Brief: This is function to Master write address byte with write request

Parameters

address	uint8 address to select a certain address

Returns

uint8 true when finished

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.11.2.8 I2C_Master_write_slave_address_with_read_req()

```
uint8 I2C_Master_write_slave_address_with_read_req (
            uint8 address )
```

Brief: This is function to Master write address byte with read request

Parameters

address	uint8 address to select a certain address

Returns

uint8 true when finished

References I2C_Wait().

Referenced by EEPROM_Read().

4.11.2.9 I2C_Master_write_byte()

```
uint8 I2C_Master_write_byte (
            uint8 data )
```

Brief: This is function to Master write data byte

Parameters

data	uint8 data to write data

Returns

uint8 true when finished

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.11.2.10 I2C_Master_read_byte()

```
uint8 I2C_Master_read_byte (
           void  )
```

Brief: This is function to Master read data byte

Parameters

void	

Returns

uint8 to get value of data

References I2C_Wait().

Referenced by EEPROM_Read().

4.12 I2C.h File Reference

I2C Module Header File for this program.

```
#include "std_types.h"
```

Functions

- void **I2C_Master_Init** (void)
  Brief: *This is the function to initialize ECU as Master Mode*
- void **I2C_Start** (void)
  Brief: *This is the function to Start I2C communication protocol*
- void **I2C_Stop** (void)
  Brief: *This is the function to Stop I2C communication protocol*
- void **I2C_Restart** (void)
  Brief: *This is the function to Restart I2C communication protocol*
- void **I2C_Wait** (void)
  Brief: *This is the I2C wait function*
- void **I2C_NAck** (void)
  Brief: *This is the I2C not Ack function*
- **uint8 I2C_Master_write_slave_address_with_write_req** (**uint8** address)
  Brief: *This is function to Master write address byte with write request*
- **uint8 I2C_Master_write_slave_address_with_read_req** (**uint8** address)
  Brief: *This is function to Master write address byte with read request*
- **uint8 I2C_Master_write_byte** (**uint8** data)
  Brief: *This is function to Master write data byte*
- **uint8 I2C_Master_read_byte** (void)
  Brief: *This is function to Master read data byte*

4.12.1 Detailed Description

I2C Module Header File for this program.

Author

Nour

Date

10/8/2020

Version

1.0

4.12.2 Function Documentation

4.12.2.1 I2C_Master_Init()

```
void I2C_Master_Init (
            void  )
```

Brief: This is the function to initialize ECU as Master Mode

Parameters

void	

Returns

void

References _XTAL_FREQ, and I2C_BAUDRATE.

Referenced by EEPROM_Init().

4.12.2.2 I2C_Start()

```
void I2C_Start (
            void  )
```

Brief: This is the function to Start I2C communication protocol

Parameters

void	

Returns

void

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.12.2.3 I2C_Stop()

```
void I2C_Stop (
            void  )
```

Brief: This is the function to Stop I2C communication protocol

Parameters

void	

Returns

void

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.12.2.4 I2C_Restart()

```
void I2C_Restart (
            void  )
```

Brief: This is the function to Restart I2C communication protocol

Parameters

void	

Returns

void

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.12.2.5 I2C_Wait()

```
void I2C_Wait (
            void  )
```

Brief: This is the I2C wait function

Parameters

void	

Returns

void

Referenced by I2C_Master_read_byte(), I2C_Master_write_byte(), I2C_Master_write_slave_address_with_read↩_req(), I2C_Master_write_slave_address_with_write_req(), I2C_NAck(), I2C_Restart(), I2C_Start(), and I2C_↩Stop().

4.12.2.6 I2C_NAck()

```
void I2C_NAck (
            void  )
```

Brief: This is the I2C not Ack function

Parameters

void	

Returns

void

References I2C_Wait().

Referenced by EEPROM_Read().

4.12.2.7 I2C_Master_write_slave_address_with_write_req()

```
uint8 I2C_Master_write_slave_address_with_write_req (
            uint8 address )
```

Brief: This is function to Master write address byte with write request

Parameters

address	uint8 address to select a certain address

Returns

uint8 true when finished

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.12.2.8 I2C_Master_write_slave_address_with_read_req()

```
uint8 I2C_Master_write_slave_address_with_read_req (
            uint8 address )
```

Brief: This is function to Master write address byte with read request

Parameters

address	uint8 address to select a certain address

Returns

uint8 true when finished

References I2C_Wait().

Referenced by EEPROM_Read().

4.12.2.9 I2C_Master_write_byte()

```
uint8 I2C_Master_write_byte (
            uint8 data )
```

Brief: This is function to Master write data byte

Parameters

data	uint8 data to write data

Returns

uint8 true when finished

References I2C_Wait().

Referenced by EEPROM_Read(), and EEPROM_Write().

4.12.2.10 I2C_Master_read_byte()

```
uint8 I2C_Master_read_byte (
           void  )
```

Brief: This is function to Master read data byte

Parameters

void	

Returns

uint8 to get value of data

References I2C_Wait().

Referenced by EEPROM_Read().

4.13 interrupts.c File Reference

interrupts Module Source File for this program

```
#include <pic16f877a.h>
#include "std_types.h"
```

Functions

- void interrupt **ISR** ()

Variables

- **uint8 time1_count_5** = 0
  counter for 5 sec
- **uint8 read_mode** = 0
  flag that indicates when to read the sensor
- **uint8 exit_mode** = 0
  flag that indicates when to exit the setting mode
- **uint8 display_mode** = 0
  flag that indicates when to enter the display mode

4.13.1 Detailed Description

interrupts Module Source File for this program

Author

Nour

Date

29/7/2020

Version

1.0

4.14 interrupts.h File Reference

interrupts Module Header File for this program

```
#include "std_types.h"
```

Variables

- **uint8 time1_count_5**
  counter for 5 sec
- **uint8 read_mode**
  flag that indicates when to read the sensor
- **uint8 exit_mode**
  flag that indicates when to exit the setting mode
- **uint8 display_mode**
  flag that indicates when to enter the display mode

4.14.1 Detailed Description

interrupts Module Header File for this program

Author

Nour

Date

29/7/2020

Version

1.0

4.15 LEDs.c File Reference

LEDs Module Source File for this program.

```
#include "PORT.h"
#include "DIO.h"
#include "LEDs.h"
```

Macros

- #define **HEATING_ELEMENT_LED_PORT PORTD_CONFIG**
  a preprocessor to define heating element led port
- #define **HEATING_ELEMENT_PORT PORTD_CONFIG**
  a preprocessor to define heating element port
- #define **COOLING_ELEMENT_PORT PORTD_CONFIG**
  a preprocessor to define cooling element port
- #define **HEATING_ELEMENT_LED_PIN PIN3**
  a preprocessor to define heating element led pin
- #define **HEATING_ELEMENT_PIN PIN4**
  a preprocessor to define heating element pin
- #define **COOLING_ELEMENT_PIN PIN5**
  a preprocessor to define cooling element pin

Functions

- void **LEDs_init** (void)
  Brief: *This is The LEDs Module Initialization Function*
- void **Turn_on_LED** (LEDs led)
  Brief: *This is the Function to turn on the LED*
- void **Turn_off_LED** (LEDs led)
  Brief: *This is the Function to turn off the LED*
- void **blink_LED** (void)
  Brief: *This is the Function to blink the LED*

4.15.1 Detailed Description

LEDs Module Source File for this program.

Author

Nour

Date

16/8/2020

Version

1.0

4.15.2 Function Documentation

4.15.2.1 LEDs_init()

```
void LEDs_init (
            void  )
```

Brief: This is The LEDs Module Initialization Function

Parameters

void	

Returns

void

References COOLING_ELEMENT_PIN, COOLING_ELEMENT_PORT, HEATING_ELEMENT_LED_PIN, HEATING_ELEMENT_LED_PORT, HEATING_ELEMENT_PIN, HEATING_ELEMENT_PORT, LOW, OUTPUT, set_pin_direction(), and write_pin().

Referenced by initialize_all().

4.15.2.2 Turn_on_LED()

```
void Turn_on_LED (
            LEDs led )
```

Brief: This is the Function to turn on the LED

Parameters

led	LEDs led to select led

Returns

void

4.15.2.3 Turn_off_LED()

```
void Turn_off_LED (
            LEDs led )
```

Brief: This is the Function to turn off the LED

Parameters

led	LEDs led to select led

Returns

void

4.15.2.4 blink_LED()

```
void blink_LED (
            void  )
```

Brief: This is the Function to blink the LED

Parameters

void	

Returns

void

References HEATING_ELEMENT_LED_PIN, HEATING_ELEMENT_LED_PORT, and toggle_pin().

4.16 LEDs.h File Reference

LEDs Module Header File for this program.

Enumerations

- enum **LEDs** { **Heating_Element_LED**, **Heating_Element**, **Cooling_Element** }

Functions

- void **LEDs_init** (void)
  Brief: *This is The LEDs Module Initialization Function*
- void **Turn_on_LED** (LEDs led)
  Brief: *This is the Function to turn on the LED*
- void **Turn_off_LED** (LEDs led)
  Brief: *This is the Function to turn off the LED*
- void **blink_LED** (void)
  Brief: *This is the Function to blink the LED*

4.16.1 Detailed Description

LEDs Module Header File for this program.

Author

Nour

Date

16/8/2020

Version

1.0

4.16.2 Function Documentation

4.16.2.1 LEDs_init()

```
void LEDs_init (
            void  )
```

Brief: This is The LEDs Module Initialization Function

Parameters

void	

Returns

void

References COOLING_ELEMENT_PIN, COOLING_ELEMENT_PORT, HEATING_ELEMENT_LED_PIN, HEATI↩NG_ELEMENT_LED_PORT, HEATING_ELEMENT_PIN, HEATING_ELEMENT_PORT, LOW, OUTPUT, set_pin↩_direction(), and write_pin().

Referenced by initialize_all().

4.16.2.2 Turn_on_LED()

```
void Turn_on_LED (
            LEDs led )
```

Brief: This is the Function to turn on the LED

Parameters

led	LEDs led to select led

Returns

void

4.16.2.3 Turn_off_LED()

```
void Turn_off_LED (
            LEDs led )
```

Brief: This is the Function to turn off the LED

Parameters

led	LEDs led to select led

Returns

void

4.16.2.4 blink_LED()

```
void blink_LED (
            void  )
```

Brief: This is the Function to blink the LED

Parameters

void	

Returns

void

References HEATING_ELEMENT_LED_PIN, HEATING_ELEMENT_LED_PORT, and toggle_pin().

4.17 LM35_temp_sensor.c File Reference

LM35_temp_sensor Module Source File for this program.

```
#include "ADC.h"
#include "LM35_temp_sensor.h"
```

Functions

- void **LM35_sensor_init** (void)

  Brief: *This is LM35_temp_sensor Module Initialization Function*
- **uint8 LM35_sensor_reading** (void)

  Brief: *This is function to read the LM35 sensor*

4.17.1 Detailed Description

LM35_temp_sensor Module Source File for this program.

Author

Nour

Date

17/8/2020

Version

1.0

4.17.2 Function Documentation

4.17.2.1 LM35_sensor_init()

```
void LM35_sensor_init (
            void  )
```

Brief: This is LM35_temp_sensor Module Initialization Function

Parameters

void	

Returns

void

Referenced by initialize_all().

4.17.2.2 LM35_sensor_reading()

```
uint8 LM35_sensor_reading (
            void  )
```

Brief: This is function to read the LM35 sensor

Parameters

void	

Returns

uint8 to get value of LM35 sensor

References ADC_readChannel().

4.18 LM35_temp_sensor.h File Reference

LM35_temp_sensor Module Header File for this program.

```
#include "std_types.h"
```

Functions

- void **LM35_sensor_init** (void)
  Brief: *This is LM35_temp_sensor Module Initialization Function*
- **uint8 LM35_sensor_reading** (void)
  Brief: *This is function to read the LM35 sensor*

4.18.1 Detailed Description

LM35_temp_sensor Module Header File for this program.

Author

Nour

Date

17/8/2020

Version

1.0

4.18.2 Function Documentation

4.18.2.1 LM35_sensor_init()

```
void LM35_sensor_init (
            void  )
```

Brief: This is LM35_temp_sensor Module Initialization Function

Parameters

void	

Returns

void

Referenced by initialize_all().

4.18.2.2 LM35_sensor_reading()

```
uint8 LM35_sensor_reading (
            void  )
```

Brief: This is function to read the LM35 sensor

Parameters

void	

Returns

uint8 to get value of LM35 sensor

References ADC_readChannel().

4.19 main.c File Reference

main Module Source File for this program

```
#include <xc.h>
#include "App.h"
#include "Buttons.h"
```

Functions

- void **main** (void)

4.19.1 Detailed Description

main Module Source File for this program

Author

Nour

Date

27/8/2020

Version

1.0

4.20 PIN_config.h File Reference

PIN_config Module Header File for this program.

Macros

- #define **PIN0** 0x01
  a preprocessor to define pin0
- #define **PIN1** 0x02
  a preprocessor to define pin1
- #define **PIN2** 0x04
  a preprocessor to define pin2
- #define **PIN3** 0x08
  a preprocessor to define pin3
- #define **PIN4** 0x10
  a preprocessor to define pin4
- #define **PIN5** 0x20
  a preprocessor to define pin5
- #define **PIN6** 0x40
  a preprocessor to define pin6
- #define **PIN7** 0x80
  a preprocessor to define pin7
- #define **FIRST_GROUP** 0x0F
  a preprocessor to define first group
- #define **SECOND_GROUP** 0xF0
  a preprocessor to define second group

4.20.1 Detailed Description

PIN_config Module Header File for this program.

Author

Nour

Date

27/7/2020

Version

1.0

4.21 PORT.c File Reference

PORT Module Source File for this program.

```
#include "PORT.h"
#include <pic16f877a.h>
```

Functions

- void **set_pin_direction** (**uint8** port, **uint8** pin, **uint8** direction)

  Brief: *This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)*
- void **set_port_direction** (**uint8** port, **uint8** direction)

  Brief: *This is a function to select certain port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)*
- void **set_group_direction** (**uint8** port, **uint8** group, **uint8** direction)

  Brief: *This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)*

4.21.1 Detailed Description

PORT Module Source File for this program.

Author

Nour

Date

27/7/2020

Version

1.0

4.21.2 Function Documentation

4.21.2.1 set_pin_direction()

```
void set_pin_direction (
              uint8 port,
              uint8 pin,
              uint8 direction )
```

Brief: This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)

Parameters

port	uint8 port to select port
pin	uint8 pin to select pin
direction	uint8 direction to select direction

Returns

void

References INPUT, OUTPUT, PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by ADC_readChannel(), Buttons_init(), LEDs_init(), and seven_segment_init().

4.21.2.2 set_port_direction()

```
void set_port_direction (
              uint8 port,
              uint8 direction )
```

Brief: This is a function to select certain port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)

Parameters

port	uint8 port to select port
direction	uint8 direction to select direction

Returns

void

References INPUT, OUTPUT, PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by seven_segment_init().

4.21.2.3 set_group_direction()

```
void set_group_direction (
              uint8 port,
              uint8 group,
              uint8 direction )
```

Brief: This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)

Parameters

port	uint8 port to select port
group	uint8 group to select group
direction	uint8 direction to select direction

Returns

void

References INPUT, OUTPUT, PORTB_CONFIG, PORTC_CONFIG, and PORTD_CONFIG.

Referenced by seven_segment_init().

4.22 PORT.h File Reference

PORT Module Header File for this program.

```
#include "std_types.h"
#include "PORT_config.h"
#include "PIN_config.h"
```

Functions

- void **set_pin_direction** (**uint8** port, **uint8** pin, **uint8** direction)

  Brief: *This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)*
- void **set_port_direction** (**uint8** port, **uint8** direction)

  Brief: *This is a function to select certain port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)*
- void **set_group_direction** (**uint8** port, **uint8** group, **uint8** direction)

  Brief: *This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)*

4.22.1 Detailed Description

PORT Module Header File for this program.

Author

Nour

Date

27/7/2020

Version

1.0

4.22.2 Function Documentation

4.22.2.1 set_pin_direction()

```
void set_pin_direction (
              uint8 port,
              uint8 pin,
              uint8 direction )
```

Brief: This is a function to select certain pin of pins (0->7) of port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)

Parameters

port	uint8 port to select port
pin	uint8 pin to select pin
direction	uint8 direction to select direction

Returns

void

References INPUT, OUTPUT, PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and P↩ ORTE_CONFIG.

Referenced by ADC_readChannel(), Buttons_init(), LEDs_init(), and seven_segment_init().

4.22.2.2 set_port_direction()

```
void set_port_direction (
              uint8 port,
              uint8 direction )
```

Brief: This is a function to select certain port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)

Parameters

port	uint8 port to select port
direction	uint8 direction to select direction

Returns

void

References INPUT, OUTPUT, PORTA_CONFIG, PORTB_CONFIG, PORTC_CONFIG, PORTD_CONFIG, and PORTE_CONFIG.

Referenced by seven_segment_init().

4.22.2.3 set_group_direction()

```
void set_group_direction (
              uint8 port,
              uint8 group,
              uint8 direction )
```

Brief: This is a function to select certain group of groups (FIRST_GROUP , SECOND_GROUP) of port of ports (A,B,C,D,E) to set the direction (INPUT , OUTPUT)

Parameters

port	uint8 port to select port
group	uint8 group to select group
direction	uint8 direction to select direction

Returns

void

References INPUT, OUTPUT, PORTB_CONFIG, PORTC_CONFIG, and PORTD_CONFIG.

Referenced by seven_segment_init().

4.23 PORT_config.h File Reference

PORT_config Module Header File for this program.

Macros

- #define **PORTA_CONFIG** 1

  a preprocessor to define portA
- #define **PORTB_CONFIG** 2

  a preprocessor to define portB
- #define **PORTC_CONFIG** 3

  a preprocessor to define portC
- #define **PORTD_CONFIG** 4

  a preprocessor to define portD
- #define **PORTE_CONFIG** 5

  a preprocessor to define portE

4.23.1 Detailed Description

PORT_config Module Header File for this program.

Author

Nour

Date

27/7/2020

Version

1.0

4.24 seven_segments.c File Reference

seven_segments Module Source File for this program

```
#include <htc.h>
#include "PORT.h"
#include "DIO.h"
#include "seven_segments.h"
```

Macros

- #define **_XTAL_FREQ** 4000000
- #define **CONTROL**

  a preprocessor to define control mode
- #define **BCD**

  a preprocessor to define BCD usage
- #define **SEG_7_DIR PORTB_CONFIG**

  a preprocessor to define seven segments direction
- #define **SEC_7_GROUP FIRST_GROUP**

  a preprocessor to define seven segments group
- #define **RIGHT_SEVEN_SEGMENT_PORT PORTD_CONFIG**

  a preprocessor to define right seven segments port
- #define **LEFT_SEVEN_SEGMENT_PORT PORTD_CONFIG**

  a preprocessor to define left seven segments port
- #define **RIGHT_SEVEN_SEGMENT_PIN PIN6**

  a preprocessor to define right seven segments pin
- #define **LEFT_SEVEN_SEGMENT_PIN PIN7**

  a preprocessor to define left seven segments pin

Functions

- void **seven_segment_init** (void)

  Brief: *This is The seven segments Module Initialization Function*
- void **display_num** (**uint8** num)

  Brief: *This is the Function to display number on 7 segment*
- void **display_num_2_7_seg** (**uint8** num)

  Brief: *This is the Function to display number on 2 7 segments*
- void **turn_off_2_7_seg** (void)

  Brief: *This is the Function to turn off 2 7 segments*

4.24.1 Detailed Description

seven_segments Module Source File for this program

Author

Nour

Date

28/7/2020

Version

1.0

4.24.2 Function Documentation

4.24.2.1 seven_segment_init()

```
void seven_segment_init (
            void  )
```

Brief: This is The seven segments Module Initialization Function

Parameters

void	

Returns

void

References LEFT_SEVEN_SEGMENT_PIN, LEFT_SEVEN_SEGMENT_PORT, LOW, OUTPUT, RIGHT_SEV↩EN_SEGMENT_PIN, RIGHT_SEVEN_SEGMENT_PORT, SEC_7_GROUP, SEG_7_DIR, set_group_direction(), set_pin_direction(), set_port_direction(), write_group(), write_pin(), and write_port().

Referenced by initialize_all().

4.24.2.2 display_num()

```
void display_num (
              uint8 num )
```

Brief: This is the Function to display number on 7 segment

Parameters

num	uint8 num to set num

Returns

void

References SEC_7_GROUP, SEG_7_DIR, write_group_value(), and write_port().

Referenced by display_num_2_7_seg().

4.24.2.3 display_num_2_7_seg()

```
void display_num_2_7_seg (
              uint8 num )
```

Brief: This is the Function to display number on 2 7 segments

Parameters

num	uint8 num to set num

Returns

void

References display_num(), HIGH, LEFT_SEVEN_SEGMENT_PIN, LEFT_SEVEN_SEGMENT_PORT, LOW, RIGHT_SEVEN_SEGMENT_PIN, RIGHT_SEVEN_SEGMENT_PORT, and write_pin().

4.24.2.4 turn_off_2_7_seg()

```
void turn_off_2_7_seg (
            void  )
```

Brief: This is the Function to turn off 2 7 segments

Parameters

void	

Returns

void

References LEFT_SEVEN_SEGMENT_PIN, LEFT_SEVEN_SEGMENT_PORT, LOW, RIGHT_SEVEN_SEGMENT_PIN, RIGHT_SEVEN_SEGMENT_PORT, and write_pin().

4.25 seven_segments.h File Reference

seven_segments Module Header File for this program

```
#include "std_types.h"
```

Functions

- void **seven_segment_init** (void)

  Brief: *This is The seven segments Module Initialization Function*
- void **display_num** (**uint8** num)

  Brief: *This is the Function to display number on 7 segment*
- void **display_num_2_7_seg** (**uint8** num)

  Brief: *This is the Function to display number on 2 7 segments*
- void **turn_off_2_7_seg** (void)

  Brief: *This is the Function to turn off 2 7 segments*

4.25.1 Detailed Description

seven_segments Module Header File for this program

Author

Nour

Date

28/7/2020

Version

1.0

4.25.2 Function Documentation

4.25.2.1 seven_segment_init()

```
void seven_segment_init (
            void  )
```

Brief: This is The seven segments Module Initialization Function

Parameters

void	

Returns

void

References LEFT_SEVEN_SEGMENT_PIN, LEFT_SEVEN_SEGMENT_PORT, LOW, OUTPUT, RIGHT_SEV↩EN_SEGMENT_PIN, RIGHT_SEVEN_SEGMENT_PORT, SEC_7_GROUP, SEG_7_DIR, set_group_direction(), set_pin_direction(), set_port_direction(), write_group(), write_pin(), and write_port().

Referenced by initialize_all().

4.25.2.2 display_num()

```
void display_num (
              uint8 num )
```

Brief: This is the Function to display number on 7 segment

Parameters

num	uint8 num to set num

Returns

void

References SEC_7_GROUP, SEG_7_DIR, write_group_value(), and write_port().

Referenced by display_num_2_7_seg().

4.25.2.3 display_num_2_7_seg()

```
void display_num_2_7_seg (
              uint8 num )
```

Brief: This is the Function to display number on 2 7 segments

Parameters

num	uint8 num to set num

Returns

void

References display_num(), HIGH, LEFT_SEVEN_SEGMENT_PIN, LEFT_SEVEN_SEGMENT_PORT, LOW, RIGHT_SEVEN_SEGMENT_PIN, RIGHT_SEVEN_SEGMENT_PORT, and write_pin().

4.25.2.4 turn_off_2_7_seg()

```
void turn_off_2_7_seg (
             void  )
```

Brief: This is the Function to turn off 2 7 segments

Parameters

void	

Returns

void

References LEFT_SEVEN_SEGMENT_PIN, LEFT_SEVEN_SEGMENT_PORT, LOW, RIGHT_SEVEN_SEGME↩
NT_PIN, RIGHT_SEVEN_SEGMENT_PORT, and write_pin().

4.26 std_types.h File Reference

Standard Types Header File for this program.

Macros

- #define **LOW** 0u
  a preprocessor to define low
- #define **HIGH** 1u
  a preprocessor to define high
- #define **FALSE** 0u
  a preprocessor to define false
- #define **TRUE** 1u
  a preprocessor to define true
- #define **INPUT** 1u
  a preprocessor to define input
- #define **OUTPUT** 0u
  a preprocessor to define output
- #define **NULL_PTR** (void ∗)0
  a preprocessor to define null pointer

Typedefs

- typedef unsigned char **uint8**
  a user defined datatype to define uint8
- typedef unsigned short **uint16**
  a user defined datatype to define uint16
- typedef unsigned long **uint32**
  a user defined datatype to define uint32

4.26.1 Detailed Description

Standard Types Header File for this program.

Author

Nour

Date

16/9/2019

Version

1.0

4.27 Timer.c File Reference

Timer Module Source File for this program.

```
#include <pic16f877a.h>
#include "Timer.h"
#include "LEDs.h"
```

Functions

- void **Timer_init** (**Timer_ConfigType** ∗Config_Ptr)

  Brief: *This is Timer Module Initialization Function*
- void **Turn_on_timer** (Timer timer)

  Brief: *This is function to turn on the timer*
- void **Turn_off_timer** (Timer timer)

  Brief: *This is function to turn off the timer*

4.27.1 Detailed Description

Timer Module Source File for this program.

Author

Nour

Date

29/7/2020

Version

1.0

4.27.2 Function Documentation

4.27.2.1 Timer_init()

```
void Timer_init (
              Timer_ConfigType * Config_Ptr )
```

Brief: This is Timer Module Initialization Function

Parameters

Config_Ptr	**Timer_ConfigType** (p. **??**) ∗ Config_Ptr to select Config_Ptr

Returns

void

4.27.2.2 Turn_on_timer()

```
void Turn_on_timer (
            Timer timer )
```

Brief: This is function to turn on the timer

Parameters

timer	Timer timer to select timer

Returns

void

4.27.2.3 Turn_off_timer()

```
void Turn_off_timer (
            Timer timer )
```

Brief: This is function to turn off the timer

Parameters

timer	Timer timer to select timer

Returns

void

4.28 Timer.h File Reference

Timer Module Header File for this program.

```
#include "std_types.h"
```

Data Structures

- struct **Timer_ConfigType**

Enumerations

- enum **Timer** { **Timer0**, **Timer1**, **Timer2** }
- enum **Timer0_Prescalar** {
  PRS_2_0, **PRS_4_0**, **PRS_8_0**, **PRS_16_0**,
  PRS_32_0, **PRS_64_0**, **PRS_128_0**, **PRS_256_0** }
- enum **Timer1_Prescalar** { **PRS_1_1**, **PRS_2_1**, **PRS_4_1**, **PRS_8_1** }
- enum **Timer2_Prescalar** { **PRS_1_2**, **PRS_4_2**, **PRS_16_2** }
- enum **Timer0_CLK** { **Internal**, **T0CKI** }
- enum **Timer1_CLK** { **internal**, **T1CKI** }

Functions

Timer_ConfigType

Timer_ConfigType *(*p. **??***) responsible for dynamic configuration of timer module*

- void **Timer_init** (**Timer_ConfigType** ∗Config_Ptr)
  Brief: *This is Timer Module Initialization Function*
- void **Turn_on_timer** (Timer timer)
  Brief: *This is function to turn on the timer*
- void **Turn_off_timer** (Timer timer)
  Brief: *This is function to turn off the timer*

4.28.1 Detailed Description

Timer Module Header File for this program.

Author

Nour

Date

29/7/2020

Version

1.0

4.28.2 Function Documentation

4.28.2.1 Timer_init()

```
void Timer_init (
            Timer_ConfigType * Config_Ptr )
```

Brief: This is Timer Module Initialization Function

Parameters

Config_Ptr	**Timer_ConfigType** (p. **??**) ∗ Config_Ptr to select Config_Ptr

Returns

void

4.28.2.2 Turn_on_timer()

```
void Turn_on_timer (
            Timer timer )
```

Brief: This is function to turn on the timer

Parameters

timer	Timer timer to select timer

Returns

void

4.28.2.3 Turn_off_timer()

```
void Turn_off_timer (
            Timer timer )
```

Brief: This is function to turn off the timer

Parameters

timer	Timer timer to select timer

Returns

void