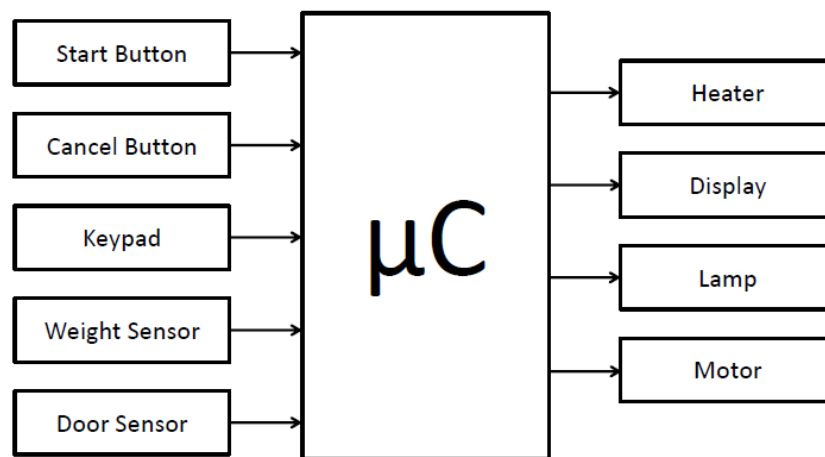


Simple Microwave

Project Idea

This project demonstrates the idea of a simple microwave, the project is designed based on layered software architecture concept and a state machine was designed.

System Diagram

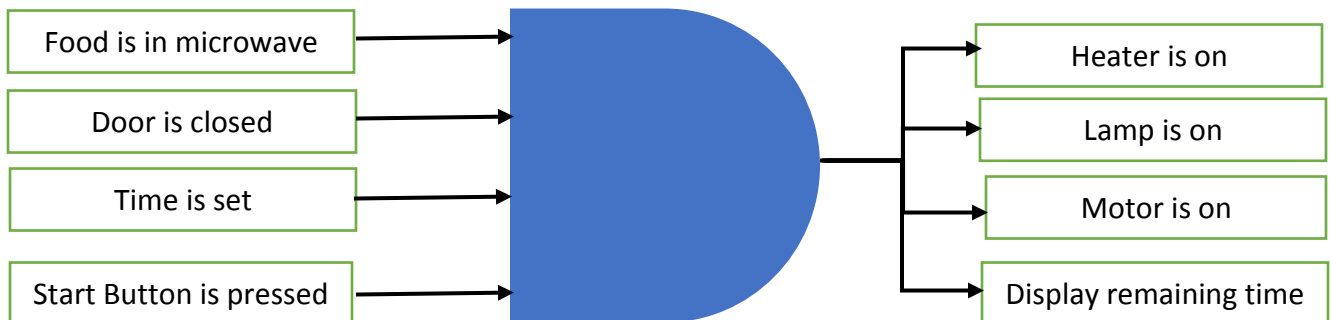


Specifications

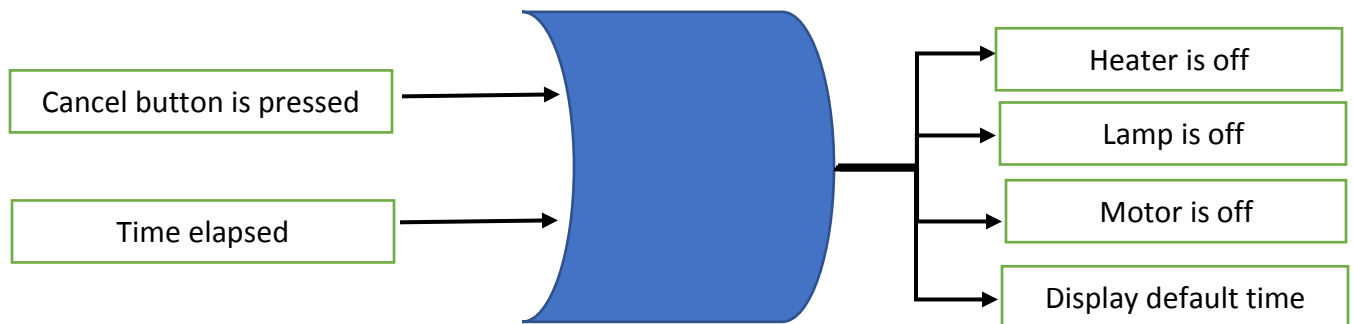
- Keypad is used to enter the time of heating only when the food is inside the microwave and the door is closed.
- LCD displays time setting: "00:00" as default or if microwave is not heating.
- LCD displays time setting when food is in the microwave and door is closed.
- LCD displays time remaining on operation state.
- User is required to enter seconds and/or minutes, maximum 99 each.
- Heater and Lamp are represented as LEDs.
- Display is represented as LCD.
- Start button, cancel button, weight Sensor and door Sensor are represented as push buttons.

States

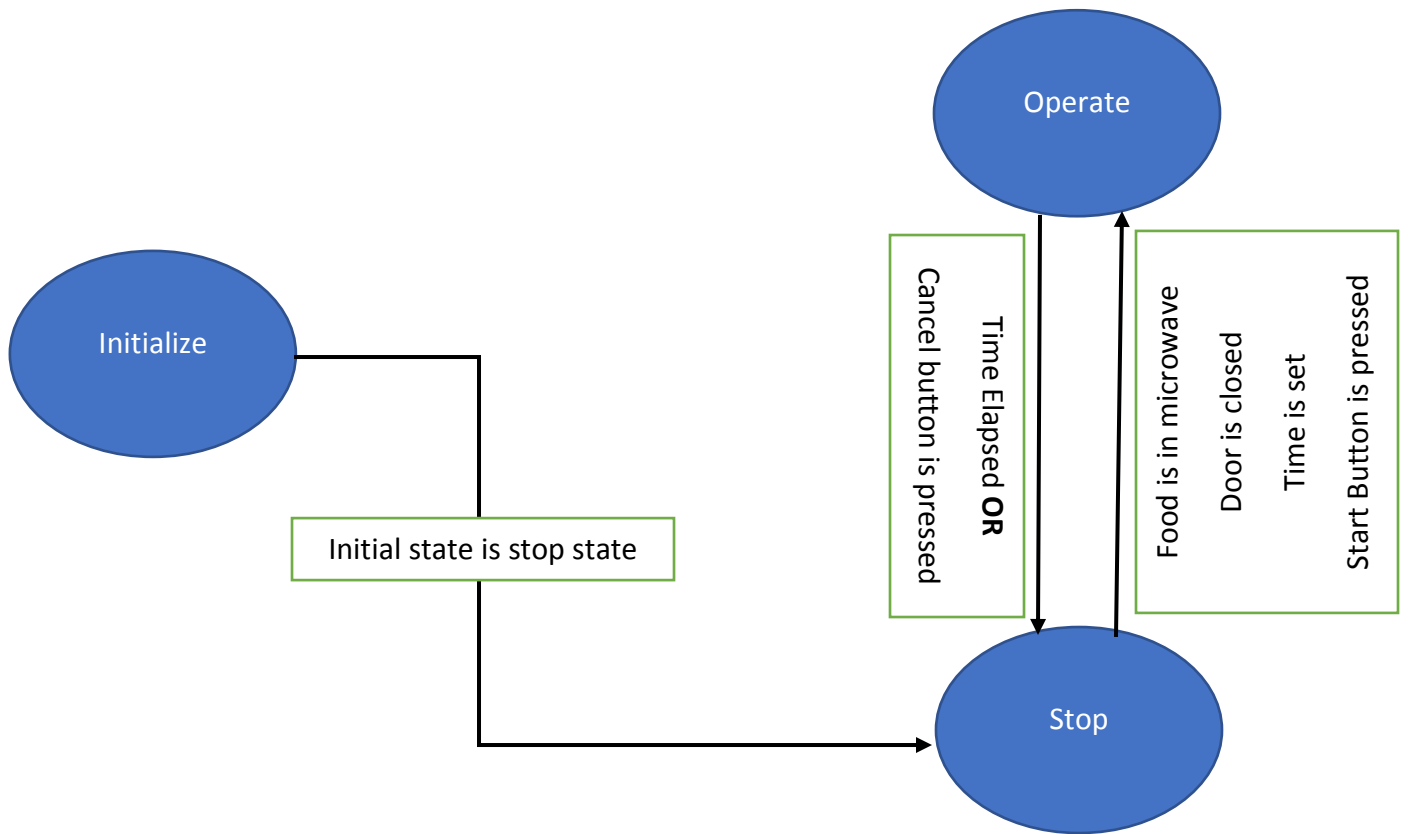
Operation State



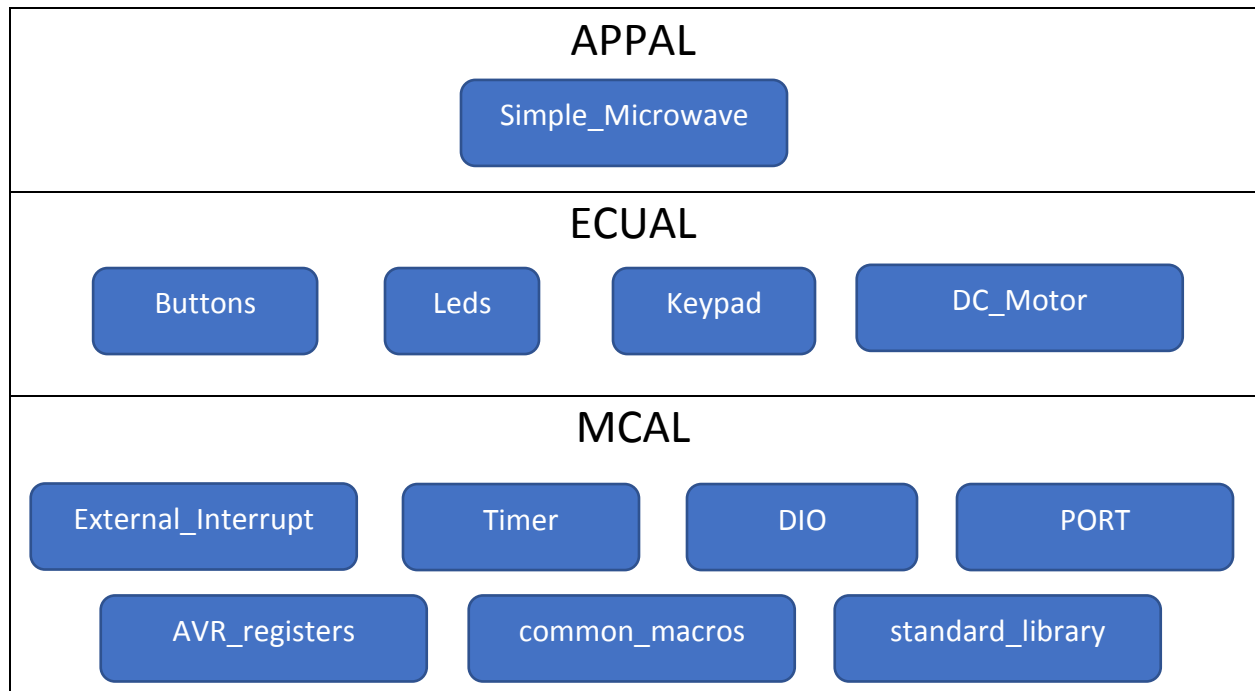
Stop State



State Diagram



Layered Architecture



APPAL

Simple_Microwave Module

Function	Type
void init_all(void)	Initialization
void Timer_App(void)	Global periodic function
void operate_all(void)	Global periodic function
void stop_all(void)	Global periodic function
void set_time(void)	Global periodic function

Init_all function

Function prototype	void init_all(void)
Description	Initialize all
Input	None
Output	None
Return	Void

Timer_App function

Function prototype	void Timer_App(void)
Description	Call back function called by timer interrupt
Input	None
Output	None
Return	Void

operate_all function

Function prototype	void operate_all(void)
Description	Operation state
Input	None
Output	None
Return	Void

stop_all function

Function prototype	void stop_all(void)
Description	Stop state
Input	None
Output	None
Return	Void

set_time function

Function prototype	void set_time (void)
Description	Set the time of the microwave
Input	None
Output	None
Return	Void

ECUAL

Leds Module

Function	Type
void leds_init(void)	Initialization

leds_init function

Function prototype	void leds_init (void)
Description	Initialization leds
Input	None
Output	None
Return	Void

Buttons Module

Function	Type
void buttons_init(void)	Initialization
void weight_sensor_API(void)	Local periodic function
void door_sensor_API(void)	Local periodic function
void cancel_button_API(void)	Local periodic function

buttons_init function

Function prototype	void buttons_init (void)
Description	Initialization buttons
Input	None
Output	None
Return	Void

weight_sensor_API function

Function prototype	void weight_sensor_API(void)
Description	Function called when external interrupt corresponding to weight sensor initiated
Input	None
Output	None
Return	Void

door_sensor_API function

Function prototype	void door_sensor_API(void)
Description	Function called when external interrupt corresponding to door sensor initiated
Input	None
Output	None
Return	Void

cancel_button_API function

Function prototype	void cancel_button_API(void)
Description	Function called when external interrupt corresponding to cancel button initiated
Input	None
Output	None
Return	Void

keypad Module

Function	Type
uint8 Keypad_getPressedKey(void)	Global periodic function
uint8 KeyPad_4x3_adjustKeyNumber(uint8 button_number)	Local periodic function
uint8 KeyPad_4x4_adjustKeyNumber(uint8 button_number)	Local periodic function

Keypad_getPressedKey function

Function prototype	uint8 Keypad_getPressedKey(void)
Description	Get the value of keypad pressed button
Input	None
Output	uint8 refers to keypad input number
Return	uint8

KeyPad_4x3_adjustKeyNumber function

Function prototype	uint8 KeyPad_4x3_adjustKeyNumber(uint8 button_number)
Description	Adjust button number of 4x3 keypad
Input	uint8 refers to keypad input number
Output	uint8 refers to adjusted value
Return	uint8

KeyPad_4x4_adjustKeyNumber function

Function prototype	uint8 KeyPad_4x4_adjustKeyNumber(uint8 button_number)
Description	Adjust button number of 4x4 keypad
Input	uint8 refers to keypad input number
Output	uint8 refers to adjusted value
Return	uint8

DC_Motor Module

Function	Type
void DC_Motor_init(void)	Initialization
void DC_Motor_rotate_clockwise(void)	Global periodic function
void DC_Motor_rotate_anti_clockwise(void)	Global periodic function
void DC_Motor_stop(void)	Global periodic function

DC_Motor_init function

Function prototype	void DC_Motor_init(void)
Description	Initialization DC motor
Input	None
Output	None
Return	void

DC_Motor_rotate_clockwise function

Function prototype	void DC_Motor_rotate_clockwise(void)
Description	Rotate DC motor clockwise
Input	None
Output	None
Return	void

DC_Motor_rotate_anti_clockwise function

Function prototype	void DC_Motor_rotate_anti_clockwise (void)
Description	Rotate DC motor anti_clockwise
Input	None
Output	None
Return	void

DC_Motor_stop function

Function prototype	void DC_Motor_stop (void)
Description	Stop DC motor
Input	None
Output	None
Return	void

MCAL

External_Interrupt Module

Function	Type
void Interrupt_init(Interrupt_ConfigType * Config_Ptr)	Initialization
void Interrupt0_DeInit(void)	Deinitialization
void Interrupt1_DeInit(void)	Deinitialization
void Interrupt2_DeInit(void)	Deinitialization
void Interrupt0_setEdgeDetectionType(State state)	Global periodic function
void Interrupt1_setEdgeDetectionType(State state)	Global periodic function
void Interrupt2_setEdgeDetectionType(State state)	Global periodic function
void INT0_setCallBack(void(*a_ptr)(void))	Global function called once
void INT1_setCallBack(void(*a_ptr)(void))	Global function called once
void INT2_setCallBack(void(*a_ptr)(void))	Global function called once
ISR(INT0_vect)	Interrupt service routine
ISR(INT1_vect)	Interrupt service routine
ISR(INT2_vect)	Interrupt service routine

Interrupt_init function

Function prototype	void Interrupt_init(Interrupt_ConfigType * Config_Ptr)
Description	Initialization external interrupts
Input	Interrupt_ConfigType represents pointer to interrupt configuration structure
Output	None
Return	void

Interrupt0_DeInit function

Function prototype	void Interrupt0_DeInit(void)
Description	De-initialize interrupt0
Input	None
Output	None
Return	void

Interrupt1_DeInit function

Function prototype	void Interrupt1_DeInit(void)
Description	De-initialize interrupt1
Input	None
Output	None
Return	void

Interrupt2_DeInit function

Function prototype	void Interrupt2_DeInit(void)
Description	De-initialize interrupt2
Input	None
Output	None
Return	void

Interrupt0_setEdgeDetectionType function

Function prototype	void Interrupt0_setEdgeDetectionType(State state)
Description	set edge of external interrupt0
Input	State refers to rising or falling edge
Output	None
Return	void

Interrupt1_setEdgeDetectionType function

Function prototype	void Interrupt1_setEdgeDetectionType(State state)
Description	set edge of external interrupt1
Input	State refers to rising or falling edge
Output	None
Return	void

Interrupt2_setEdgeDetectionType function

Function prototype	void Interrupt2_setEdgeDetectionType(State state)
Description	set edge of external interrupt2
Input	State refers to rising or falling edge
Output	None
Return	void

INT0_setCallBack function

Function prototype	void INT0_setCallBack(void(*a_ptr)(void))
Description	call back function when interrupt0 is initiated
Input	void(*a_ptr)(void) refers to function called by the call-back function in upper layer
Output	None
Return	void

INT1_setCallBack function

Function prototype	<code>void INT1_setCallBack(void(*a_ptr)(void))</code>
Description	call back function when interrupt1 is initiated
Input	<code>void(*a_ptr)(void)</code> refers to function called by the call-back function in upper layer
Output	None
Return	void

INT2_setCallBack function

Function prototype	<code>void INT2_setCallBack(void(*a_ptr)(void))</code>
Description	call back function when interrupt2 is initiated
Input	<code>void(*a_ptr)(void)</code> refers to function called by the call-back function in upper layer
Output	None
Return	void

Timer Module

Function	Type
void Timer_init(Timer_ConfigType * Config_Ptr)	Initialization
void Timer0_DeInit(void)	Deinitialization
void Timer1_DeInit(void)	Deinitialization
void Timer2_DeInit(void)	Deinitialization
void Timer0_clearTimerValue(void)	Global periodic function
void Timer1_clearTimerValue(void)	Global periodic function
void Timer2_clearTimerValue(void)	Global periodic function
uint8 Timer0_getTimerValue(void)	Global periodic function
Uint16 Timer1_getTimerValue(void)	Global periodic function
uint8 Timer2_getTimerValue(void)	Global periodic function
void Timer0_OVF_setInterrupt(void)	Global periodic function
void Timer0_COMP_setInterrupt(void)	Global periodic function
void Timer1_OVF_setInterrupt(void)	Global periodic function
void Timer1_COMPA_setInterrupt(void)	Global periodic function
void Timer1_COMPB_setInterrupt(void)	Global periodic function
void Timer2_OVF_setInterrupt(void)	Global periodic function
void Timer2_COMP_setInterrupt(void)	Global periodic function
void Timer0_OVF_clearInterrupt(void)	Global periodic function
void Timer0_COMP_clearInterrupt(void)	Global periodic function
void Timer1_OVF_clearInterrupt(void)	Global periodic function
void Timer1_COMPA_clearInterrupt(void)	Global periodic function
void Timer1_COMPB_clearInterrupt(void)	Global periodic function
void Timer2_OVF_clearInterrupt(void)	Global periodic function
void Timer2_COMP_clearInterrupt(void)	Global periodic function
void Timer0_OVF_setCallBack(void(*a_ptr)(void))	Global function called once
void Timer0_COMP_setCallBack(void(*a_ptr)(void))	Global function called once
void Timer1_OVF_setCallBack(void(*a_ptr)(void))	Global function called once
void Timer1_COMPA_setCallBack(void(*a_ptr)(void))	Global function called once
void Timer1_COMPB_setCallBack(void(*a_ptr)(void))	Global function called once

void Timer2_OVF_setCallBack(void(*a_ptr)(void))	Global function called once
void Timer2_COMP_setCallBack(void(*a_ptr)(void))	Global function called once
ISR(TIMERO_OVF_vect)	Interrupt service routine
ISR(TIMERO_COMP_vect)	Interrupt service routine
ISR(TIMER1_OVF_vect)	Interrupt service routine
ISR(TIMER1_COMPA_vect)	Interrupt service routine
ISR(TIMER1_COMPB_vect)	Interrupt service routine
ISR(TIMER2_OVF_vect)	Interrupt service routine
ISR(TIMER2_COMP_vect)	Interrupt service routine

Timer_init function

Function prototype	void Timer_init(Timer_ConfigType * Config_Ptr)
Description	Initialization timers
Input	Interrupt_ConfigType represents pointer to timer configuration structure
Output	None
Return	void

Timer0_Delnit function

Function prototype	void Timer0_Delnit(void)
Description	De-initialize timer0
Input	None
Output	None
Return	void

Timer1_Delnit function

Function prototype	void Timer1_Delnit(void)
Description	De-initialize timer1
Input	None
Output	None
Return	void

Timer2_DeInit function

Function prototype	void Timer2_DeInit(void)
Description	De-initialize timer2
Input	None
Output	None
Return	void

Timer0_clearTimerValue function

Function prototype	void Timer0_clearTimerValue (void)
Description	Clear timer0 value
Input	None
Output	None
Return	void

Timer1_clearTimerValue function

Function prototype	void Timer1_clearTimerValue (void)
Description	Clear timer1 value
Input	None
Output	None
Return	void

Timer2_clearTimerValue function

Function prototype	void Timer2_clearTimerValue (void)
Description	Clear timer2 value
Input	None
Output	None
Return	void

Timer0_getTimerValue function

Function prototype	uint8 Timer0_getTimerValue(void)
Description	Get timer0 value
Input	None
Output	uint8 represents timer0 value
Return	uint8

Timer1_getTimerValue function

Function prototype	uint16 Timer1_getTimerValue(void)
Description	Get timer1 value
Input	None
Output	uint16 represents timer0 value
Return	uint16

Timer2_getTimerValue function

Function prototype	uint8 Timer2_getTimerValue(void)
Description	Get timer2 value
Input	None
Output	uint8 represents timer0 value
Return	uint8

Timer0_OVF_setInterrupt function

Function prototype	void Timer0_OVF_setInterrupt(void)
Description	Set interrupt for timer0 overflow mode
Input	None
Output	None
Return	void

Timer0_COMP_setInterrupt function

Function prototype	void Timer0_COMP_setInterrupt(void)
Description	Set interrupt for timer0 CTC mode
Input	None
Output	None
Return	void

Timer1_OVF_setInterrupt function

Function prototype	void Timer1_OVF_setInterrupt(void)
Description	Set interrupt for timer1 overflow mode
Input	None
Output	None
Return	void

Timer1_COMPA_setInterrupt function

Function prototype	void Timer1_COMPA_setInterrupt(void)
Description	Set interrupt for timer1 CTC mode channelA
Input	None
Output	None
Return	void

Timer1_COMPB_setInterrupt function

Function prototype	void Timer1_COMPB_setInterrupt(void)
Description	Set interrupt for timer1 CTC mode channelB
Input	None
Output	None
Return	void

Timer2_OVF_setInterrupt function

Function prototype	void Timer2_OVF_setInterrupt(void)
Description	Set interrupt for timer2 overflow mode
Input	None
Output	None
Return	void

Timer2_COMP_setInterrupt function

Function prototype	void Timer2_COMP_setInterrupt(void)
Description	Set interrupt for timer2 CTC mode
Input	None
Output	None
Return	void

Timer0_OVF_clearInterrupt function

Function prototype	void Timer0_OVF_clearInterrupt(void)
Description	Clear interrupt for timer0 overflow mode
Input	None
Output	None
Return	void

Timer0_COMP_clearInterrupt function

Function prototype	void Timer0_COMP_clearInterrupt(void)
Description	Clear interrupt for timer0 CTC mode
Input	None
Output	None
Return	void

Timer1_OVF_clearInterrupt function

Function prototype	void Timer1_OVF_clearInterrupt(void)
Description	Clear interrupt for timer1 overflow mode
Input	None
Output	None
Return	void

Timer1_COMPA_clearInterrupt function

Function prototype	void Timer1_COMPA_clearInterrupt(void)
Description	Clear interrupt for timer1 CTC mode channelA
Input	None
Output	None
Return	void

Timer1_COMPB_clearInterrupt function

Function prototype	void Timer1_COMPB_clearInterrupt(void)
Description	Clear interrupt for timer1 CTC mode channelB
Input	None
Output	None
Return	void

Timer2_OVF_clearInterrupt function

Function prototype	void Timer2_OVF_clearInterrupt(void)
Description	Clear interrupt for timer2 overflow mode
Input	None
Output	None
Return	void

Timer2_COMP_clearInterrupt function

Function prototype	void Timer2_COMP_clearInterrupt(void)
Description	Clear interrupt for timer2 CTC mode
Input	None
Output	None
Return	void

Timer0_OVF_setCallBack function

Function prototype	void Timer0_OVF_setCallBack(void(*a_ptr)(void))
Description	call back function when timer0 overflow interrupt is initiated
Input	void(*a_ptr)(void) refers to function called by the call-back function in upper layer
Output	None
Return	void

Timer0_COMP_setCallBack function

Function prototype	void Timer0_COMP_setCallBack(void(*a_ptr)(void))
Description	call back function when timer0 CTC interrupt is initiated
Input	void(*a_ptr)(void) refers to function called by the call-back function in upper layer
Output	None
Return	void

Timer1_OVF_setCallBack function

Function prototype	void Timer1_OVF_setCallBack(void(*a_ptr)(void))
Description	call back function when timer1 overflow interrupt is initiated
Input	void(*a_ptr)(void) refers to function called by the call-back function in upper layer
Output	None
Return	void

Timer1_COMPA_setCallBack function

Function prototype	void Timer1_COMPA_setCallBack(void(*a_ptr)(void))
Description	call back function when timer1 channelA CTC interrupt is initiated
Input	void(*a_ptr)(void) refers to function called by the call-back function in upper layer
Output	None
Return	void

Timer1_COMPB_setCallBack function

Function prototype	void Timer1_COMPB_setCallBack(void(*a_ptr)(void))
Description	call back function when timer1 channelB CTC interrupt is initiated
Input	void(*a_ptr)(void) refers to function called by the call-back function in upper layer
Output	None
Return	void

Timer2_OVF_setCallBack function

Function prototype	void Timer2_OVF_setCallBack(void(*a_ptr)(void))
Description	call back function when timer2 overflow interrupt is initiated
Input	void(*a_ptr)(void) refers to function called by the call-back function in upper layer
Output	None
Return	void

Timer2_COMP_setCallBack function

Function prototype	void Timer2_COMP_setCallBack(void(*a_ptr)(void))
Description	call back function when timer2 CTC interrupt is initiated
Input	void(*a_ptr)(void) refers to function called by the call-back function in upper layer
Output	None
Return	void

DIO Module

Macro	Type
write_pin(port,pin,value)	Macro replaced periodically
write_port(port,value)	Macro replaced periodically
write_group(port,group,value)	Macro replaced periodically
read_pin(port,pin)	Macro replaced periodically

write_pin macro

Macro	write_pin(port,pin,value)
Description	Write output to a certain pin
Input	port, pin and value: high or low
Output	None

write_port macro

Macro	write_port(port,value)
Description	Write output to a port
Input	port and value: high or low
Output	None

write_group macro

Macro	write_group(port,group,value)
Description	Write output to a group
Input	port, group and value: high or low
Output	None

read_pin macro

Macro	read_pin(port,pin)
Description	Read input from a pin
Input	port and pin
Output	value of the pin

Port Module

Macro	Type
set_direction(port,pin,direction)	Macro replaced once
set_direction_group(port,group,direction)	Macro replaced once
set_direction_port(port,direction)	Macro replaced once

set_direction macro

Macro	set_direction(port,pin,direction)
Description	set the direction of a pin
Input	port, pin and direction: output or input
Output	None

set_direction_group macro

Macro	set_direction_group(port,group,direction)
Description	set the direction of a group
Input	port, group and direction: output or input
Output	None

set_direction_port macro

Macro	set_direction_port(port,direction)
Description	set the direction of a port
Input	port and direction: output or input
Output	None