SOFTWARE LAYERS DESCRIPTION

The software is divided into three layers, to make it more generic:

1) Micro-controller abstraction layer:

Consists of functions that allow the initialization and configuration of the micro-controller's ports without the need to deal with the control registers directly.

2) Device drivers' layer:

Each device driver consists of a number of functions that enables the control of the external device.

3) Application layer:

The functions created in the previous two layers are used in the main, to control the devices in a manner that satisfies the project's requirements.

Micro-controller abstraction functions

- Port_Init Function:

Description: Initializes the port using its control registers.

Inputs: Port index.

Outputs: Port is initialized.

-Port_SetPinDirection:

Description: Set the direction of a specified pin in a port.

Inputs: Port index, pin mask and desired pin direction.

Outputs: The specified pin direction is set as either input or output.

-Port_SetPinPullUp:

Description: Enable the pull-up resistor for a specified pin in a port.

Inputs: Port index, pin mask and enable.

Outputs: The pull-up resistor for the specified pin is enabled.

-Port_SetPinPullDown:

Description: Enable the pull-up resistor for a desired pin.

Inputs: Port index, pin mask and enable.

Outputs: The pull-up resistor for the specified pin is enabled.

-DIO_WritePort:

Description: Change the value of the pins selected in a port.

Inputs: Port index, pin mask and the value to write to the pin.

Outputs: Value is written to pin.

-DIO_ReadPort:

Description: Get value of pin selected by pin mask in a port.

Inputs: Port index and pin mask.

Outputs: Returns value of selected pin in the port.

-DIO_FlipPort:

Description: Toggles the value of pin selected by pin mask in a port.

Inputs: Port index and pin mask.

Outputs: Value of selected pin in port selected is toggled.

Device drivers' layer

LCD driver functions

-Send_instruction:

Description: Send a command to the LCD.

Inputs: Command hexadecimal code.

Outputs: The LCD executes the command sent *e.g.0x01 will clear the LCD.*

- LCD_init:

Description: Choose whether to display 8-bits or 4-bits, prepare the screen for display and initializes its ports.

Inputs: None.

Outputs: LCD is ready to display.

- LCD_Clear:

Description: Clears LCD screen.

Inputs: None.

Outputs: LCD displays clear screen.

- LCD_DisplayChar:

Description: Displays a single character on the screen.

Inputs: Character to be displayed.

Outputs: A single character is displayed on the LCD screen.

- LCD_DisplayString:

Description: Displays a string on the screen.

Inputs: String to be displayed.

Outputs: The string is displayed on the LCD screen.

- LCD_DisplayInt:

Description: Displays an integer number on the screen.

Inputs: Integer number to be displayed.

Outputs: The number is displayed on the LCD screen.

- LCD_DisplayFloat:

Description: Displays a float number on the screen.

Inputs: Number to be displayed.

Outputs: The number is displayed on the LCD screen.

- LCD_Select_RowCol:

Description: Specify the position to display at.

Inputs: Row number and column number.

Outputs: The output is displayed at the specified coordinate.

PWM driver functions

- PWM_Gen_Init:

Description: Initializes PWM generator.

Inputs: PWM module, generator, divider, output state and load value.

Outputs: Initialize module clock, set PWM divider, set output waveform state, set load value, start timer and enable PWM output.

- PWM_Pin_Init:

Description: Initializes PWM generator and a desired pin.

Inputs: PWM module, generator, divider, output state and load value.

Outputs: Initialize module clock, set PWM divider, set output waveform state, set load value, start timer and enable PWM output.

- PWM_Write:

Description: Set duty cycle for a desired pin.

Inputs: PWM pin and duty cycle.

Outputs: The PWM pin will exhibit the specified duty cycle.

Servo driver functions

- SERVO_Init:

Description: Initializes the servo motor.

Inputs: None.

Outputs: Servo motor is initialized and ready to control.

- SERVO_Rotate:

Description: Rotate the motor either clock wise or anti-clock wise.

Inputs: Degree of rotation and the direction.

Outputs: Motor rotates in the specified direction.

UART driver functions

- UART_Init:

Description: Initializes UART module.

Inputs: None.

Outputs: UART is initialized.

- UART_InChar:

Description: Wait for new serial port input.

Inputs: None.

Outputs: ASCII code for key typed.

- UART_OutChar:

Description: Transfer letter.

Inputs: letter to be transferred.

Outputs: 8-bit to serial port.

- UART_OutString:

Description: Send string.

Inputs: pointer to null-terminated string to be transferred.

Outputs: String is transferred.

- UART_InUDec:

Description: accepts ASCII input in unsigned decimal format and converts to a 32-bit unsigned number.

Inputs: None.

Outputs: 32-bits unsigned number.

- UART_OutDec:

Description: Gives a 32-bit number in unsigned decimal format.

Inputs: 32-bit number to be transferred.

Outputs: None.

- UART_OutDec:

Description: Gives a 32-bit number in unsigned decimal format.

Inputs: 32-bit number to be transferred.

Outputs: None.

- UART_InUHex:

Description: Accepts ASCII input in unsigned hexadecimal (base 16) format.

Inputs: None.

Outputs: 32-bit unsigned number.

- UART_OutUHex:

Description: Output a 32-bit number in unsigned hexadecimal format.

Inputs: 32-bit number to be transferred.

Outputs: None.

- UART_InString:

Description: Accepts ASCII characters from the serial port and adds them to a string until <enter> is typed or until max length of the string is reached. It echoes each character as it is inputed. If a backspace is inputted, the string is modified and the backspace is echoed.

Inputs: Pointer to empty buffer, size of buffer.

Outputs: Null terminated string.

ADC driver functions

LOW LEVEL FUNCTIONS:

-ADC_module:

Description: select which ADC.

Input: which of ADC {ADC0, ADC1}.

Output: it is a void function it just chooses the ADC0/ADC1 configuration.

-ADC0_selec_DIS_SEQ:

Description: select sequencer for ADC0.

Input: which of Sequencer {S0, S1, S2, S3} for ADC1.

Output: it is a void function it just chooses the SEQUENCER {S0, S1, S2, S3} configuration for ADC0.

-ADC1_selec_DIS_SEQ

Description: select sequencer for ADC1.

Input: which of Sequencer {S0, S1, S2, S3} for ADC1.

Output: it is a void function it just chooses the SEQUENCER {S0, S1, S2, S3} configuration for ADC1.

-ADC0_select_Trigger

Description: Choose the software trigger for ADC0.

Input: which of which trigger.

Output: it is a void function; it just chooses the trigger configuration for ADC0.

-ADC1_select_Trigger

Description: Choose the software trigger for ADC1.

Input: which of which trigger.

Output: it is a void function; it just chooses the trigger configuration for ADC1.

-ADC0_SEL_IN

Description: Select ADC0 input .

Input: which Input Channel for ADC0.

Output: it is a void function; it just chooses the channel configuration for ADC0.

-**ADC1_SEL_IN**

Description: Select ADC1 input.

Input: which Input Channel for ADC1.

Output: it is a void function; it just chooses the channel configuration for ADC1.

-ADC0_SET_CTL

Description: Set ADC0 Control.

Input: the control mask, which sequencer we are on.

Output: it is a void function; it just makes the control configuration for ADC0.

-ADC1_SET_CTL

Description: Set ADC1 Control

Input: the control mask, which sequencer we are on

Output: it is a void function; it just makes the control configuration for ADC1

HIGH LEVEL FUNCTIONS:

-ADC_Init

Description: Initiates the ADC driver

Input: which ADC, which Sequencer, the sequencer mask, trigger value, which channel, control mask

Output: it initializes the ADC driver

-ADC_GPIO_PIN_CONFIG

Description: Port configuration; It initializes PORTE or A OR B to use their pins as channels

Input: Port, Pin

Output: it initializes the Port driver and set the pin for ADC

-ADC0_SEQ_SEL_EN

Description: Enable Sequencer for ADC0

Input: Sequencer mask” enabling its bits”

Output: void function for setting the sequencer registers’ bits

-ADC1_SEQ_SEL_EN

Description: Enable Sequencer for ADC1

Input: Sequencer mask” enabling its bits”

Output: void function for setting the sequencer registers’ bits

-ADC0_TAKE_SAMPLE

Description: Start a new conversion, making ADC0 ready to take a new sample

Input: Sequencer mask

Output: taking a sample

-ADC1_TAKE_SAMPLE

Description: Start a new conversion, making ADC1 ready to take a new sample

Input: Sequencer mask

Output: taking a sample

-Tempreature_Value

Description: Get the Temperature Value from internal temperature Sensor

Input: register to save the value

Output: Temperature Value

-Distance Value

Description: Get the Distance Value from Potentiometer

Input: register to save the value

Output: Temperature Value