Component Design Document for DIO

**Table of Contents**

Revision History 1

*1. Introduction* 2

1.1 Objective 2

1.2 Context Diagram 2

2. *External Interface* 2

2.1 <file name.h> 2

3. *Static Design* 2

3.1 Files 2

3.2 Types 3

3.3 Symbol Define <#define> 3

3.4 Const 3

3.5 Interface (Services) 3

*4.* *Dynamic Design* 4

4.1 Mode Management 4

4.2 Sequence Diagram 4

*5.* *Shared Resources* 4

5.1 Analysis 4

5.2 Protection 4

6. *Configuration Parameters* 4

6.1 Pre-compile time 4

6.2 Link time 5

6.3 Post-build 5

*7.* *Configuration Constrains* 5

*8.* *Integration Constrains* 5

*9.* *History* 5

# *1. Introduction*

## Objective

This component is used to allow micro controller to interface with hardware layer as it is a very important part of MCAL layer which is between Microcontroller and HAL layer, we use it to configure controller pins to be used for LEDS and switches and so on we use it to read to and write pins

## Context Diagram

DIO\_ReadPin() DIO\_WritePin()

DIO\_Init()

DIO\_H

# *External Interface*

## STD\_TYPES.h

### Types

|  |  |
| --- | --- |
| Data Type | Description |
| unsigned char u8 | This is unsigned integer it’s range from 0 up to 255 |

## BIT\_MATH.h

### 2.2.1 Symbol <#defines>

SET\_BIT (var,bitno) : this macro is used to set(set it to one) certain bit it takes the u8 variable which include desired bit and number of bit that we want to set.

CLR\_BIT (var,bitno): this macro is used to clear (set it to zero)certain bit it takes the u8 variable which include desired bit and number of bit that we want to set.

# *Static Design*

## Files

### Used Files

|  |  |
| --- | --- |
| FILE NAME | Description |
| DIO.c | This file contains the definition of the APIs |
| DIO.h | This file contains the prototype of the APIs |
| DIO\_cfg.c | This file contains the configurations of the component which describe the mode of each pin as it is DefaultModeInputWZIntrnalPullup=0or  InputWithExternalPullUp or  Output. |
| DIO\_cfg.h | This file contains the configuration of components as each pin is a assigned to certain component or still ideal not used. |

### Files Inclusion

DIO\_cfg.h

DIO.h

DIO.c

DIO\_cfg.c

BIT\_MATH.h

STD\_TYPES.h

## Types

|  |  |
| --- | --- |
| Req ID | DIO\_001 |
| Covers | HLD\_001 |
| Name | DIO\_PinName\_T |
| Type | Enumeration |
| Range | 0-32 |
| Description | It’s used as index for controller pins as it’s configurable to microcontroller pins and last element is indicator to number of pins |

|  |  |
| --- | --- |
| Req ID | DIO\_001 |
| Covers | HLD\_001 |
| Name | DIO\_Mode\_t |
| Type | Enumeration |
| Range | 0-2 |
| Description | It is used to list the modes of DIO pins |

## Symbol Define <#define>

|  |  |
| --- | --- |
| Req ID | DIO\_001 |
| Covers | HLD\_001 |
| Name | DDRA |
| Type | U8 |
| Range | 0-255 |
| Description | Used to access data direction register of PORTA |

|  |  |
| --- | --- |
| Req ID | DIO\_002 |
| Covers | HLD\_001 |
| Name | DDRB |
| Type | U8 |
| Range | 0-255 |
| Description | Used to access data direction register of PORTB |

|  |  |
| --- | --- |
| Req ID | DIO\_003 |
| Covers | HLD\_001 |
| Name | DDRC |
| Type | U8 |
| Range | 0-255 |
| Description | Used to access data direction register of PORTC |

|  |  |
| --- | --- |
| Req ID | DIO\_004 |
| Covers | HLD\_001 |
| Name | PORTA |
| Type | U8 |
| Range | 0-255 |
| Description | Used to access data register of PORTA |

|  |  |
| --- | --- |
| Req ID | DIO\_005 |
| Covers | HLD\_001 |
| Name | PORTB |
| Type | U8 |
| Range | 0-255 |
| Description | Used to access data register of PORTB |

|  |  |
| --- | --- |
| Req ID | DIO\_006 |
| Covers | HLD\_001 |
| Name | PORTC |
| Type | U8 |
| Range | 0-255 |
| Description | Used to access data register of PORTC |

|  |  |
| --- | --- |
| Req ID | DIO\_007 |
| Covers | HLD\_001 |
| Name | PINA |
| Type | U8 |
| Range | 0-255 |
| Description | Used to get data from controller PORTA pins |

|  |  |
| --- | --- |
| Req ID | DIO\_008 |
| Covers | HLD\_001 |
| Name | PINB |
| Type | U8 |
| Range | 0-255 |
| Description | Used to get data from controller PORTB pins |

|  |  |
| --- | --- |
| Req ID | DIO\_009 |
| Covers | HLD\_001 |
| Name | PINC |
| Type | U8 |
| Range | 0-255 |
| Description | Used to get data from controller PORTC pins |

## Const

|  |  |
| --- | --- |
| Req ID | DIO\_010 |
| Covers | HLD\_001 |
| Name | const DIO\_Mode\_t DIO\_Mode |
| Type | Structure |
| Range | 0-4 |
| Description | This structure to assign the mode of each pin. |

## Interface (Services)

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | DIO\_011 | | |
| Covers | HLD\_001 | | |
| Name/protoTypes | void DIO\_Int(void) | | |
| Service ID | 0x01 | | |
| Re-entrant / Non re- entrant | Non re\_entrant | | |
| Synchronous/Asynchronous | Synchronous | | |
| Return Value | | ---------- | ---------- |
| Input parameter | | ---------- | ---------- |
| Output parameter | | ---------- | ---------- |
| Input /Output Parameter | | ---------- | ---------- |

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | DIO\_012 | | |
| Covers | HLD\_001 | | |
| Name/protoTypes | void DIO\_WritePin(u8 PinNameCpy,u8 ValueCpy) | | |
| Service ID | 0x02 | | |
| Re-entrant / Non re- entrant | Non\_re\_entrant | | |
| Synchronous/Asynchronous | Synchronous | | |
| Return Value | | ---------- | ------------------- |
| Input parameter | | PinNameCpy, ValueCpy | First input is pin name which used as index to access pins, second is value desired to be written on pin |
| Output parameter | | ---------- | -------------------- |
| Input /Output Parameter | | ---------- | -------------------- |

|  |  |  |  |
| --- | --- | --- | --- |
| Req ID | DIO\_013 | | |
| Covers | HLD\_001 | | |
| Name/protoTypes | u8 DIO\_ReadPin(u8 PinNameCpy) | | |
| Service ID | 0x03 | | |
| Re-entrant / Non re- entrant | Non\_re\_entrant | | |
| Synchronous/Asynchronous | Synchronous | | |
| Return Value | | Returne\_Value | It returns the value of pin desired to read |
| Input parameter | | PinNameCpy | input is pin name which used as index to access pins |
| Output parameter | | Returne\_Value | Is value returned from pin. |
| Input /Output Parameter | | ------------------ | ------------------- |

# *Configuration Parameters*

## Pre-compile time

### #define PORTA ((u8\*)0x3B)

### #define DDRA ((u8\*)0x3A)

### #define PINA ((volatile u8\*)0x39)

### #define PORTB ((u8\*)0x38)

### #define DDRB ((u8\*)0x37)

### #define PINB ((volatile u8\*)0x36)

### #define PORTC ((u8\*)0x35)

### #define DDRC ((u8\*)0x34)

### #define PINC ((volatile u8\*)0x33)

### #define PORTD ((u8\*)0x32)

### #define DDRD ((u8\*)0x31)

### #define PIND ((volatile u8\*)0x30)

# *Configuration Constrains*

We can’t access to number of pins in our target controller,in AVR we can’t exceeds 32 pin.

# *History*

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Christine Ramsis | 18th April 2018 | Create the template | 1.0 |
| Christine Ramsis | 19th April 2018 | External interface | 1.1 |
| Christine Ramsis | 20th April 2018 | Interfaces | 1.2 |
|  |  |  |  |
|  |  |  |  |

<The changes happened in the documents>

**Appendix A: Glossary**