# Document Name:

**GPIO\_MODULE\_SWS\_V01**

# Functionality:

We have to Access the pins to write and read from them, therefore this module is responsible for:

* Dealing with the GPIO of the Targeted Hardware
* Configuring the Pins Mode (Direction, Speed & etc.)
* Writing / Setting the Pins
* Reading from the Pins

# Configuration Parameters:

1. GPIO\_LOCK\_KEY\_REGISTER

Before writing on the GPIO Port F, there is a Lock Key to be entered in the Register to get access to

PORT F.

#define GPIO\_LOCK\_KEY 0x4C4F434B

1. SYSTEM\_CLOCK\_REGISTER

You have to enable the clock of the GPIO before dealing with The GPIO, to enable the clock there is a bit

For each Port in the register, you have to set the bit of the required GPIO Port to enable the clock of the Port.

#define SYSCTL\_RCGC2\_R (\*((volatile u32 \*)0x400FE108))

# API’s:

1-

|  |  |
| --- | --- |
| **API Prototype** | Error\_S GPIO\_Init(u8 Port,u8 Pin,u8 Direction); |
| **Description** | This API is responsible for initializing the GPIO.   1. Enables The Clock Of the Specified Port 2. \*Writes LOCK\_KEY to Activate The PORT (Applied only to PORT F) 3. Initializing The Direction of The Specified Pin by Writing on the DIR\_R and PUR\_R regsiters   #define GPIO\_PORTx\_DIR\_R (\*((volatile u32 \*)0xxxxxxxxx))  #define GPIO\_PORTx\_PUR\_R (\*((volatile u32 \*)0xxxxxxxxx)) |
| **Input Parameters** | 1. U8 Port {“A”, “B”, “C”, “D”, “E”, “F”} 2. U8 Pin {PIN\_0 = 0, PIN\_1 = 1, PIN\_2 = 2, etc.} 3. U8 Direction {1 = Output ,0 = Input} |
| **Output Parameters** | 1. Error\_S {OK = 1, NOK = 0} |

2-

|  |  |
| --- | --- |
| **API Prototype** | Error\_S GPIO\_WritePin(u8 Port,u8 Pin,u8 Value); |
| **Description** | This API is responsible for writing on A specified GPIO pin.   1. Writing on the DATA\_R register Using the Value Entered   #define GPIO\_PORTx\_DATA\_R (\*((volatile u32 \*)0xxxxxxxx)) |
| **Input Parameters** | 1. U8 Port {“A”, “B”, “C”, “D”, “E”, “F”} 2. U8 Pin {PIN\_0 = 0, PIN\_1 = 1, PIN\_2 = 2, etc.} 3. U8 Value {1 = HIgh ,0 = Input} |
| **Output Parameters** | 1. Error\_S {OK = 1, NOK = 0} |

3-

|  |  |
| --- | --- |
| **API Prototype** | Error\_S GPIO\_ReadPin(u8 Port,u8 Pin,u8 \*Value); |
| **Description** | This API is responsible for writing on A specified GPIO pin.   1. Reading From the DATA\_R register   #define GPIO\_PORTx\_DATA\_R (\*((volatile u32 \*)0xxxxxxxx)) |
| **Input Parameters** | 1. U8 Port {“A”, “B”, “C”, “D”, “E”, “F”} 2. U8 Pin {PIN\_0 = 0, PIN\_1 = 1, PIN\_2 = 2, etc.} 3. U8\* Value {Ptr to hold the Value Read From The Pin} |
| **Output Parameters** | 1. Error\_S {OK = 1, NOK = 0} |