

```

/*****

Header file for Helpers

*****/

#ifndef Helpers_H
#define Helpers_H

#include "ES_Configure.h"
#include "ES_Types.h"
#include "ES_Framework.h"
#include "Definitions.h"

// Public Function Prototypes

void GPIO_Set(int port, uint8_t bits);
void GPIO_Clear(int port, uint8_t bits);
void GPIO_Init(uint8_t SYCTL, int port_base, uint8_t bits, uint8_t direction);

#endif /* Helpers_H */

void InitPWM(
    uint8_t module,
    uint8_t block,
    uint8_t generator,
    uint32_t period
);

void setPWM_value(float duty, uint8_t module, uint8_t block, uint8_t generator,
    uint32_t period);

void InitInputCapture(
    uint8_t timer_num,
    uint8_t timer_letter,
    uint8_t priority,
    uint32_t time_length
);

void InitPeriodic(
    uint8_t timer_num,
    uint8_t timer_letter,
    uint8_t priority,
    uint32_t time_length);

void clearCaptureInterrupt(
    uint8_t timer_num,
    uint8_t timer_letter,
    uint8_t priority,
    uint32_t time_length);

void clearPeriodicInterrupt(
    uint8_t timer_num,
    uint8_t timer_letter,
    uint8_t priority,
    uint32_t time_length);

void disableCaptureInterrupt(
    uint8_t timer_num,
    uint8_t timer_letter,

```

```
    uint8_t priority,  
    uint32_t time_length);  
  
void enableCaptureInterrupt(  
    uint8_t timer_num,  
    uint8_t timer_letter,  
    uint8_t priority,  
    uint32_t time_length);  
  
void disablePeriodicInterrupt(  
    uint8_t timer_num,  
    uint8_t timer_letter,  
    uint8_t priority,  
    uint32_t time_length);  
  
void enablePeriodicInterrupt(  
    uint8_t timer_num,  
    uint8_t timer_letter,  
    uint8_t priority,  
    uint32_t time_length);  
  
uint32_t captureInterrupt(  
    uint8_t timer_num,  
    uint8_t timer_letter,  
    uint8_t priority,  
    uint32_t time_length);  
  
float clamp(float X, uint8_t min, uint8_t max);
```