Objectives

- User-created Library function

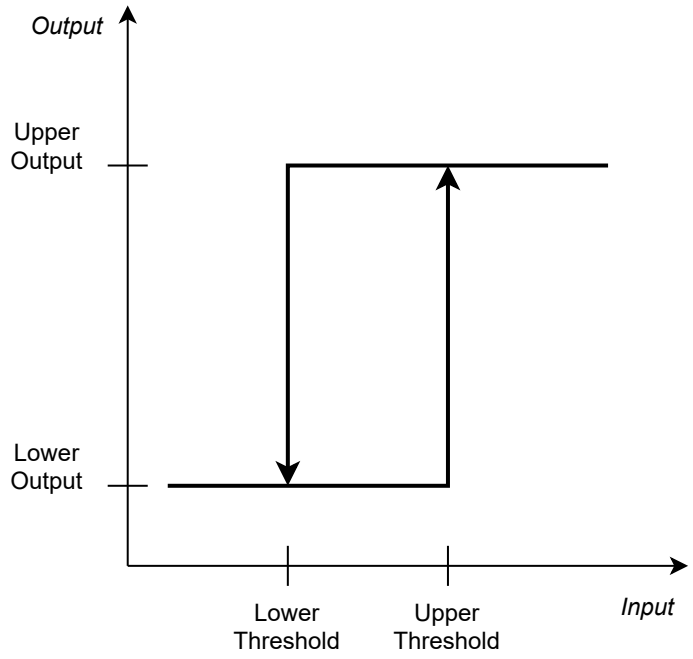
Hysteresis

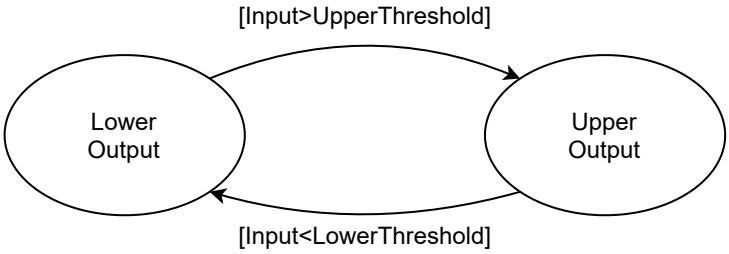
- Data type

```
typedef struct
{
    float   fltUpperThreshold;
    float   fltLowerThreshold;
    float   fltUpperOutput;
    float   fltLowerOutput;
    float   fltHystState;
} UserLib_Hyst_t;
```

- Function prototype

```
/*****************************************************************************
//*!

@brief      Hysteresis function

@param[in]   fltIn      Input argument
@param[out]  *pParam    Pointer to parameters

@return     The function returns f32 value, which represents
            the actual output value

@details

@note       All parameters and states, used by the function, can be
reset...
*****************************************************************************
***/

float UserLib_Hyst(float flt_in, UserLib_Hyst_t* );
```





Ramp

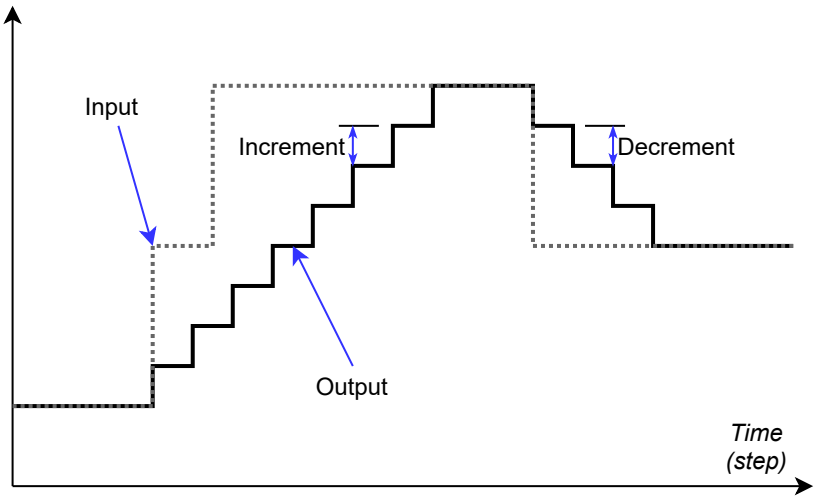
- Data Type

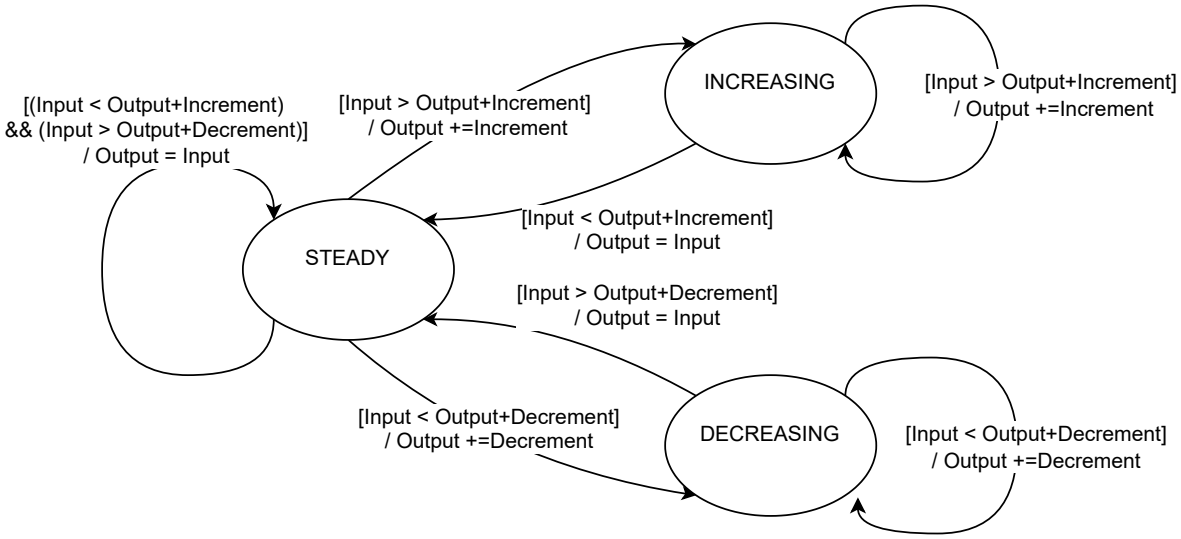
```
typedef struct
{
    float   fltState;
    float   fltIncrement;
    float   fltDecrement;
} UserLib_Ramp_t;
```

- Function Prototype

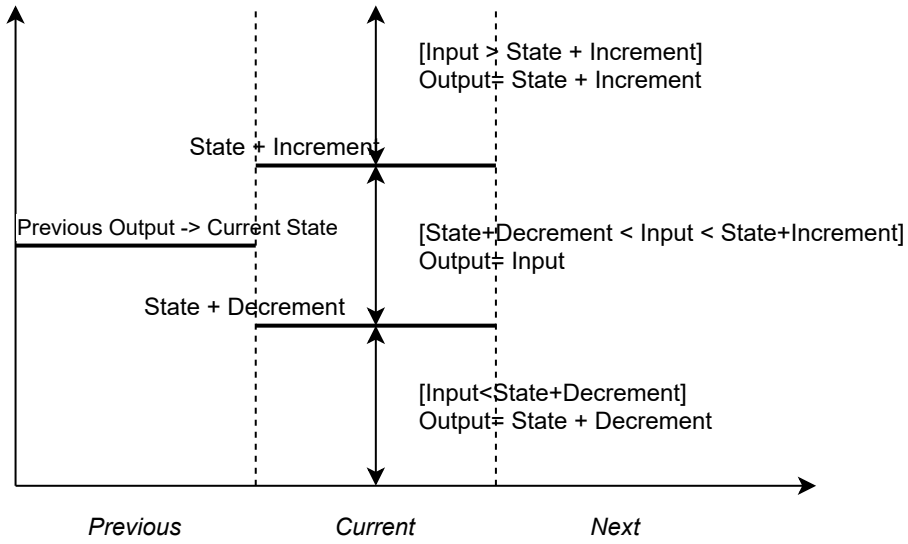
```
/*****************************************************************************
//*!

@brief      The function calculates the Increment/decrement ramp

@param[in]    fltIn      Input argument representing the desired output
value.
@param[out]  *pParam     Pointer to parameters

@return     The function returns f32 value, which represents
            the actual ramp output value

@details
```

```
12
13  @note       All parameters and states, used by the function, can be
    reset...
14  ********************************************************************************
    ***/
15  float UserLib_Ramp(float flt_in, UserLib_Ramp_t* );
16
```





`float_t RampStateMachine(float_t fltIn)`



`float_t Ramp(float_t fltIn)`