# MECHANICAL DISPLAY DEVELOPER GUIDE:

## NEW DISPLAY MODULES CREATING STEPS:

1. Set timer to 100kHz. Parameters example for 72MHz internal clock:
    a. Clock source: internal clock (72Mhz),
    b. Prescaler (PSC) = 71,
    c. Counter Period (ARR) = 9,
2. Set global interrupts in your timer (or TIM update interrupts for special timers),
3. Set 4 motor push pull output pins , 1 signal input no pull-up/no pull down pin to detecting hall sensor digital signal and check that uart pins have been set correctly,
4. Check that uart interrupts are correctly set,
5. In *main.cpp* file:
    a. Define display resolution:

```
#define SEGMENTS_PER_DISPLAY 7
```

    b. Define DisplayModule pointer:

```
DisplayModule* module1 = nullptr;
DisplayModule* module2 = nullptr;
```

    c. Init DisplayModule class (after peripheral init):

```
module1 = new DisplayModule(&htim1, SEGMENTS_PER_DISPLAY, HALL1_GPIO_Port,
HALL1_Pin, IN1_GPIO_Port,   IN1_Pin,   IN2_GPIO_Port,   IN2_Pin, IN3_GPIO_Port,
IN3_Pin,   IN4_GPIO_Port,   IN4_Pin);
module2 = new DisplayModule(&htim2, SEGMENTS_PER_DISPLAY, HALL2_GPIO_Port,
HALL2_Pin, IN2_1_GPIO_Port, IN2_1_Pin, IN2_2_GPIO_Port, IN2_2_Pin, IN2_3_GPIO_Port,
IN2_3_Pin, IN2_4_GPIO_Port, IN2_4_Pin);
```

    d. Map DisplayModule (remember that you have to map them to string sequentially “D3”, “D4” etc. because of uart commands handling):

```
displays_map["D1"] = module1;
displays_map["D2"] = module2;
```

    e. Add uart irq init:

```
HAL_UART_Receive_IT(&huart1, (uint8_t*)&single_byte, sizeof(single_byte));
```

6. In *my_tims_irq.cpp* file:
    a. Extern DisplayModule pointer:
    b. Add new callback handling:
7. In *stm32fxxx_it.c* file extern callback:

```
extern void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim);
```

8. In *uart.handler.cpp* and in *uart.handler.hpp* check that you are using correct uart.

# COMMAND STRUCTURE:

| | STRUCTURE | DISPLAY | COMMAND | PARAMETER |
|---|---|---|---|---|
| Move display/displays to target segment | **DISPLAY:COMMAND:PARAMETER** | **DX**<br><br>X - display number.<br><br>DA - all displays | **move** or **MOVE** | **S,D**<br><br>S - segment number,<br><br>D - “+” or “-” describe direction |
| Jump display/displays to next/previous segment | | | **jump** or **JUMP** | **D**<br><br>D - “+” or “-”describe direction |
| Reset display/displays to zero position (works only in one direction) | **DISPLAY:COMMAND** | | **reset** or **RESET** | |

# COMMANDS EXAMPLES:

| COMMAND | COMMAND DESCRIPTION |
|---|---|
| D1:move:3,+ | Display 1. move to segment 3 incrementally. |
| DA:jump:- | All displays jump to previous segment. |
| D4:reset | Display 4. go to zero position. |

# BLACKPILL PINOUT: