

```

typedef struct {
    UART_HandleTypeDef *uartPort;

    uint8_t buffer[11];

    float accel_x, accel_y, accel_z;
    float omega_x, omega_y, omega_z;
    float theta_x, theta_y, theta_z;

    uint32_t lastUpdated;
} JY62_HandleTypeDef;

```

```

typedef struct {
    TIM_HandleTypeDef *counter;
    TIM_HandleTypeDef *posTimer, *negTimer;
    uint32_t pos_channel, neg_channel;           // TIM_CHANNEL_n
    float kp, ki, kd;
    float dt;           // Feedback Control Period; used to perform the
calculation
    // uint16_t maxCount;           // divider used to normalize the
error; just assume it is 16 bit
    // uint16_t timerPeriod;       // multiplier used to convert
the normalized output into timer output; get this from 'timer'
    uint32_t lastTick;
    float lastError;
    float lastSpeed, last5Speed;
    float sumError;
    float goalSpeed;
} Motor_HandleTypeDef;

```

```

typedef struct {
    uint8_t x, y;
}Coordinate;

```

```

typedef struct {
    Coordinate coord;
    uint8_t isAlly;
} Beacon;

```

```

typedef struct {

```

```
    Coordinate startCoord, destCoord;
    uint32_t timeLimit;
    uint8_t reward;
    uint32_t startTime;
} Order;

typedef struct{
    Coordinate coord1, coord2;
} Obstacle;
```