**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;