Client: +Socket

- server_addr: struct sockaddr

+ Client(): Client
+ Client(int port, char *address, int domain, int type, int protocol): Client
+ Client(int port, uint32_t address, int domain, int type, int protocol): Client
+ ~Client()
+ send(void *data, size_t length): int
+ receive(void *buffer, size_t length): int

Server: +Socket

- client_addr: struct sockaddr *
- new_Socket: int
- client_socklen: int

+ Server(): Server
+ Server(int port, char *address, int domain, int type, int protocol): Server
+ Server(int port, uint32_t address, int domain, int type, int protocol): Server
+ ~Server()
+ startServer(): int
+ send(void *data, size_t length): int
+ receive(void *buffer, size_t length): int

Extends

Extends

Socket

- addr: struct sockaddr_in*
- socketfd: int
- socklen: int
- domain: int
- type: int
- protocol: int
- port: int

+ Socket(): Socket
+ Socket(int port, char *address, int domain, int type, int protocol): Socket
+ Socket(int port, uint32_t address, int domain, int type, int protocol): Socket
+ ~Socket():
+ setPort(int port): int
+ setHost(const char *host): int
+ setType(int type): int
+ setDomain(int domain): int

Accelerometer: +ICM_20948

- data_pin: int
- clock_pin: int
- i2c_handle: int
- i2c_addr: int
- accX: int16_t
- accY: int16_t
- accZ: int16_t
- gyroX: int16_t
- gyroY: int16_t
- gyroZ: int16_t

+ Accelerometer(): Accelerometer
+ Accelerometer(int i2c_addr, int data_pin, int clock_pin): Accelerometer
+ ~Accelerometer()
+ getGyroX(): int+ getGyroY(): int
+ getGyroZ(): int
+ getAccX(): int16_t
+ getAccY(): int16_t
+ getAccZ(): int16_t
+ getGyroXYZ(): signed short [3]
+ getAccXYZ(): signed short [3]

Motor

- speed: int
- range: int
- frequency: int
- pin: unsigned int

+ Motor(int pin): Motor
+Motor(int pin, int frequency): Motor
+ Motor(int pin, int frequency, int pwm_range):Motor
+ ~Motor()
+ getPin(): int
+ getFrequency(): int
+ getDutyRange(): int
+ getSpeed(): int
+ setFrequency(int frequency)
+ setSpeed(int speed): int
+ setDutyRange(int range):
+ stop(): int

Drone

- motors: Motor**
- acc: Accelerometer *
- server: Server *
- acc_data: signed short *
- gyro_data: signed short *
- hover: thread*

- calibrated: int

+ Drone: Drone
+ Drone(int port, int freq, int m1_pin, int m2_pin, int m3_pin, int m4_pin, int acc_da_pin, int acc_plk_pin, int cc): Drone
+ ~Drone()
+ handleInstruction(char Ins):int
+ checkAlt(): void *
+ getGyroData(): signed short *
+ getAccData(): signed short *
+ setAllMotors(int speed): int
+ setMotorSpeed(int motor_id, int speed):int
+ startDrone():
- getAvgMotorSpeed(): int

CONSTANTS

ACC_ADR: i2c address of accelerometer
MOTORx_PIN: Motor GPIO pin for x (x := [0, 3])
HOVER: calculated throttle % for hover