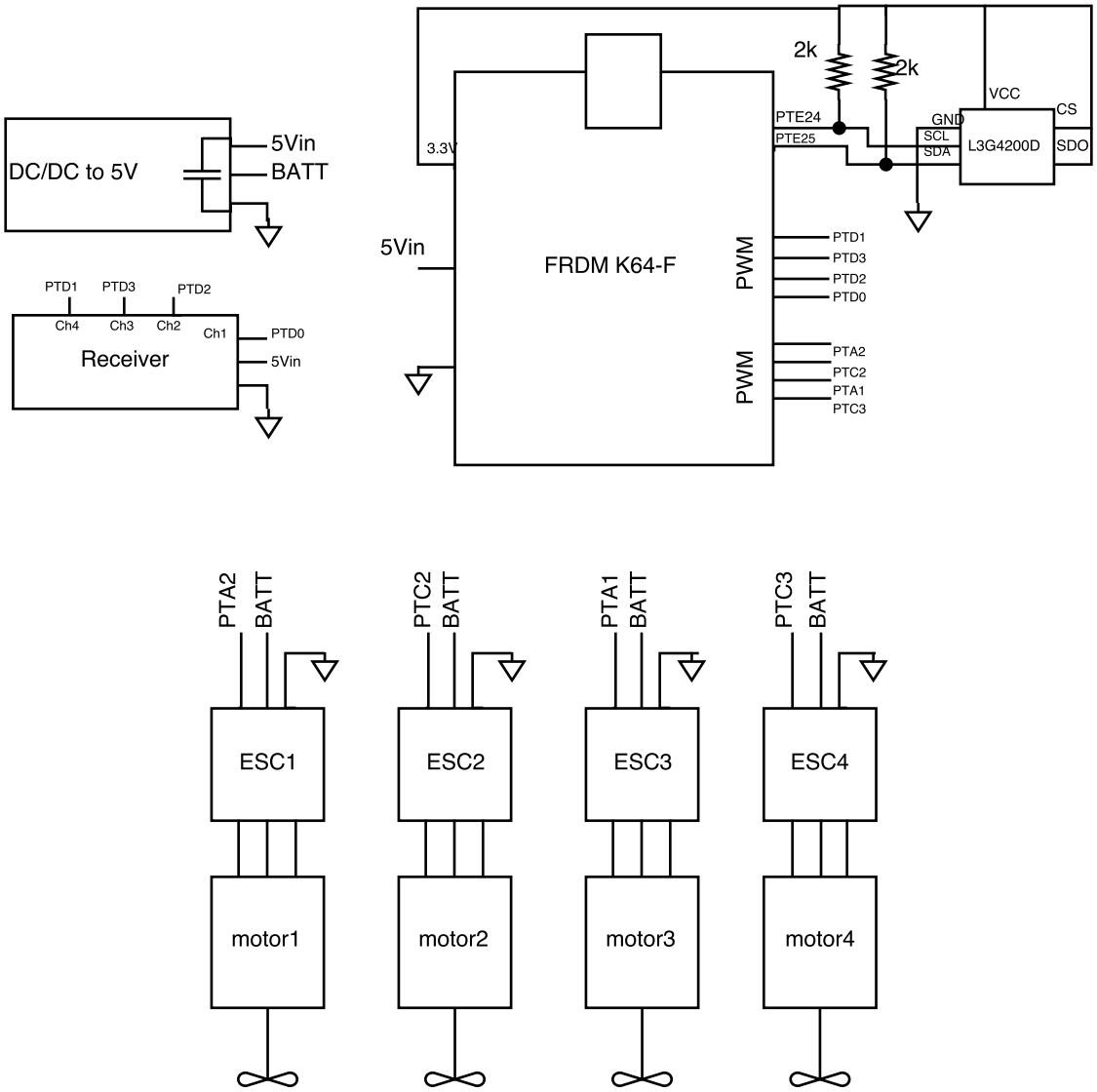
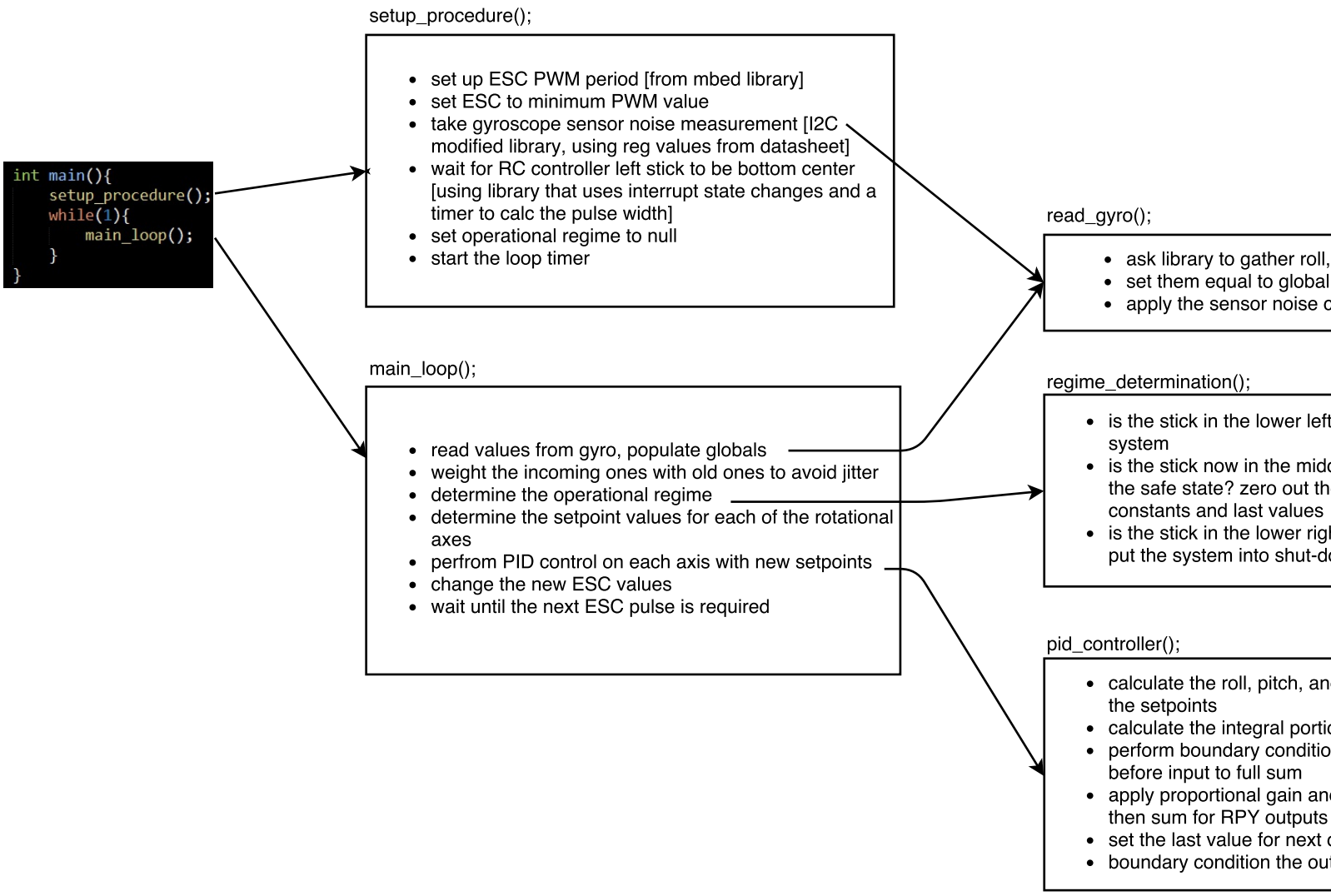
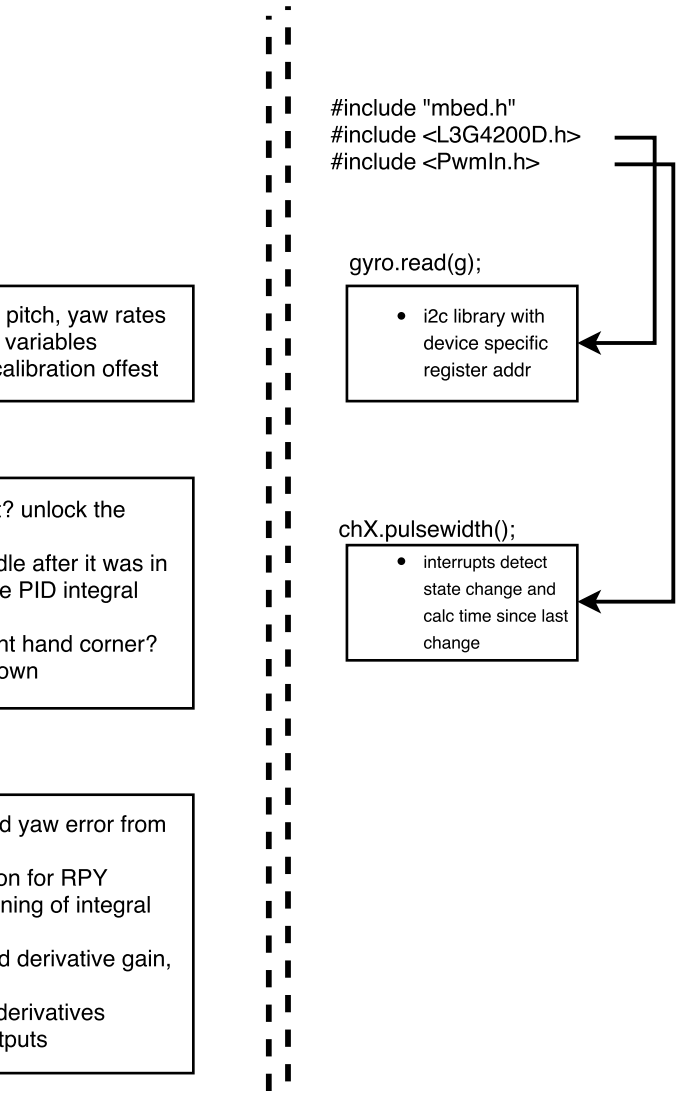

DC/DC to 5V
5Vin
BATT
PTD1
PTD3
PTD2
Ch4
Ch3
Ch2
Ch1
Receiver
PTD0
5Vin
3.3V
5Vin
FRDM K64-F
PWM
PWM
2k
2k
VCC
CS
PTE24
PTE25
GND
SCL
SDA
L3G4200D
SDO
PTD1
PTD3
PTD2
PTD0
PTA2
PTC2
PTA1
PTC3
PTA2
BATT
PTC2
BATT
PTA1
BATT
PTC3
BATT
ESC1
ESC2
ESC3
ESC4
motor1
motor2
motor3
motor4


int main(){
setup_procedure();
while(1){
main_loop();
}
}
setup_procedure();
set up ESC PWM period [from mbed library]
set ESC to minimum PWM value
take gyroscope sensor noise measurement [I2C modified library, using reg values from datasheet]
wait for RC controller left stick to be bottom center [using library that uses interrupt state changes and a timer to calc the pulse width]
set operational regime to null
start the loop timer
main_loop();
read values from gyro, populate globals
weight the incoming ones with old ones to avoid jitter
determine the operational regime
determine the setpoint values for each of the rotational axes
perfrom PID control on each axis with new setpoints
change the new ESC values
wait until the next ESC pulse is required
read_gyro();
ask library to gather roll,
set them equal to global
apply the sensor noise c
regime_determination();
is the stick in the lower left system
is the stick now in the midd the safe state? zero out th constants and last values
is the stick in the lower righ put the system into shut-d
pid_controller();
calculate the roll, pitch, an the setpoints
calculate the integral portio
perform boundary conditio before input to full sum
apply proportional gain an then sum for RPY outputs
set the last value for next
boundary condition the ou


#include "mbed.h"
#include <L3G4200D.h>
#include <PwmIn.h>
gyro.read(g);
i2c library with device specific register addr
chX.pulsewidth();
interrupts detect state change and calc time since last change
pitch, yaw rates
variables
alibration offest
? unlock the
dle after it was in
e PID integral
nt hand corner?
own
d yaw error from
on for RPY
ning of integral
d derivative gain,
derivatives
tputs