```
#include<EEPROM.h>
                                                 int cal int, start, gyro address;
void setup(){
                                                 int receiver input[5];
 pinMode(13,OUTPUT);
                                                 int temperature;
 for(int i=0;i<EEPROM.length();i++){
                                                 int acc axis[4], gyro axis[4];
                                                 float roll level adjust, pitch level adjust;
  EEPROM.write(i,0);
digitalWrite(13,HIGH);
                                                 long acc x, acc y, acc z, acc total vector;
                                                 unsigned long timer channel 1,
                                                 timer channel 2, timer channel 3,
void loop(){}
^^^^^
                                                 timer channel 4, esc timer,
                                                 esc loop timer;
#include <Wire.h>
#include <EEPROM.h>
                                                 unsigned long timer 1, timer 2, timer 3,
float pid p gain roll = 1.3;
                                                 timer 4, current time;
float pid_i_gain_roll = 0.04;
                                                 unsigned long loop timer;
float pid d gain roll = 18.0;
                                                 double gyro pitch, gyro roll, gyro yaw;
int pid max roll = 400;
                                                 double gyro axis cal[4];
float pid p gain pitch = pid p gain roll;
                                                 float pid error temp;
float pid i gain pitch = pid i gain roll;
                                                 float pid i mem roll, pid roll setpoint,
float pid d gain pitch = pid d gain roll;
                                                 gyro roll input, pid output roll,
int pid max pitch = pid max roll;
                                                 pid last roll d error;
float pid p gain yaw = 4.0;
                                                 float pid i mem pitch, pid pitch setpoint,
float pid i gain yaw = 0.02;
                                                 gyro pitch input, pid output pitch,
float pid d gain yaw = 0.0;
                                                 pid last pitch d error;
int pid max yaw = 400;
                                                 float pid i mem yaw, pid yaw setpoint,
boolean auto level = true;
                                                 gyro yaw input, pid output yaw,
byte last channel 1, last channel 2,
                                                 pid last yaw d error;
last channel 3, last channel 4;
                                                 float angle roll acc, angle pitch acc,
byte eeprom data[36];
                                                 angle pitch, angle roll;
byte highByte, lowByte;
                                                 boolean gyro angles set;
volatile int receiver input channel 1,
                                                 void setup(){
                                                  for(start = 0; start \leq 35;
receiver input channel 2,
receiver input channel 3,
                                                 start++)eeprom data[start] =
receiver input channel 4;
                                                 EEPROM.read(start);
int counter channel 1, counter channel 2,
                                                  start = 0;
counter channel 3, counter channel 4,
                                                  gyro address = eeprom data[32];
loop counter;
                                                  Wire.begin();
int esc 1, esc 2, esc 3, esc 4;
                                                  TWBR = 12;
int throttle, battery voltage;
                                                  DDRD = B11110000;
```

```
DDRB = B00110000;
                                                   PCMSK0 = (1 \leq PCINT3);
 digitalWrite(12,HIGH);
                                                   while(receiver input channel 3 < 990 ||
 while(eeprom data[33]!='J'||
                                                 receiver input channel 3 > 1020 \parallel
eeprom data[34] != 'M' || eeprom data[35]
                                                 receiver input channel 4 < 1400){
!= 'B') delay(10);
                                                    receiver input channel 3 =
 if(eeprom data[31] == 2 \parallel
                                                 convert receiver channel(3);
eeprom data[31] == 3)delay(10);
                                                    receiver input channel 4 =
                                                 convert_receiver_channel(4);
 set gyro registers();
 for (cal int = 0; cal int < 1250; cal int
                                                    start ++;
                                                    PORTD |= B11110000;
++){
  PORTD |= B11110000;
                                                    delayMicroseconds(1000);
  delayMicroseconds(1000);
                                                    PORTD &= B00001111;
  PORTD &= B00001111;
                                                    delay(3);
  delayMicroseconds(3000);
                                                    if(start == 125){
                                                     digitalWrite(12, !digitalRead(12));
 for (cal int = 0; cal int < 2000; cal int
                                                     start = 0;
++){
  if(cal int % 15 == 0)digitalWrite(12,
!digitalRead(12);
                                                   start = 0;
  gyro signalen();
                                                  battery voltage = (analogRead(0) + 65) *
  gyro axis cal[1] += gyro axis[1];
                                                  1.2317;
  gyro_axis_cal[2] += gyro axis[2];
                                                  loop timer = micros();
  gyro_axis_cal[3] += gyro_axis[3];
                                                  digitalWrite(12,LOW);
                                                 void loop(){
  //We don't want the esc's to be beeping
annoyingly. So let's give them a 1000us puls
                                                   gyro roll input = (gyro roll input * 0.7) +
                                                 ((gyro roll / 65.5) * 0.3);
while calibrating the gyro.
  PORTD |= B11110000;
                                                   gyro pitch input = (gyro pitch input *
  delayMicroseconds(1000);
                                                 0.7) + ((gyro pitch / 65.5) * 0.3);
  PORTD &= B00001111;
                                                 gyro yaw input = (gyro yaw input *0.7) +
                                                 ((gyro\ yaw / 65.5) * 0.3);
  delay(3);
                                                   angle pitch += gyro pitch * 0.0000611;
                                                   angle roll += gyro roll * 0.0000611;
 gyro axis cal[1] /= 2000;
                                                   angle_pitch -= angle_roll * sin(gyro_yaw *
 gyro axis cal[2] /= 2000;
 gyro axis cal[3] /= 2000;
                                                 0.00001066);
                                                   angle roll += angle pitch * sin(gyro yaw
 PCICR |= (1 << PCIE0);
                                                 * 0.00001066);
 PCMSK0 |= (1 << PCINT0);
 PCMSK0 |= (1 << PCINT1);
 PCMSK0 |= (1 << PCINT2);
```

```
acc total vector =
                                                    pid last yaw d error = 0;
sqrt((acc_x*acc_x)+(acc_y*acc_y)+(acc_z*
                                                   if(start == 2 && receiver input channel 3
acc z);
                                                  < 1050 && receiver input channel 4 >
 if(abs(acc y) < acc total vector){
                                                  1950)start = 0;
  angle pitch acc =
asin((float)acc_y/acc_total_vector)* 57.296;
                                                   pid roll setpoint = 0;
                                                   if(receiver input channel 1 >
                                                  1508)pid roll setpoint =
 if(abs(acc x) < acc total vector){
  angle roll acc =
                                                  receiver input channel 1 - 1508;
asin((float)acc x/acc total vector)* -
                                                   else if(receiver input channel 1 <
                                                  1492)pid roll setpoint =
57.296;
                                                  receiver input channel 1 - 1492;
 }
 angle pitch acc = 0.0;
 angle roll acc = 0.0;
                                                   pid roll setpoint -= roll level adjust;
 angle pitch = angle pitch * 0.9996 +
                                                   pid roll setpoint /= 3.0;
angle pitch acc * 0.0004;
                                                   pid pitch setpoint = 0;
 angle roll = angle roll * 0.9996 +
                                                   if(receiver input channel 2>
                                                  1508)pid pitch setpoint =
angle roll acc * 0.0004;
 pitch level adjust = angle pitch * 15;
                                                  receiver input channel 2 - 1508;
                                                   else if(receiver_input_channel_2 <
 roll_level_adjust = angle roll * 15;
 if(!auto level){
                                                  1492)pid pitch setpoint =
  pitch_level_adjust = 0;
                                                  receiver input channel 2 - 1492;
  roll level adjust = 0;
                                                   pid pitch setpoint -= pitch level adjust;
 if(receiver input channel 3 < 1050 &&
                                                   pid pitch setpoint /= 3.0;
receiver input channel 4 < 1050) start = 1;
                                                   pid yaw setpoint = 0;
 if(start == 1 && receiver input channel 3
                                                   if(receiver input channel 3 > 1050){ //Do
< 1050 && receiver input channel 4>
                                                  not yaw when turning off the motors.
1450){
                                                    if(receiver input channel 4>
                                                  1508)pid yaw setpoint =
  start = 2;
                                                  (receiver input channel 4 - 1508)/3.0;
  angle pitch = angle pitch acc;
  angle roll = angle roll acc;
                                                    else if(receiver input channel 4 <
                                                  1492)pid yaw setpoint =
  gyro_angles_set = true;
  pid i mem roll = 0;
                                                  (receiver input channel 4 - 1492)/3.0;
  pid last roll d error = 0;
  pid i mem pitch = 0;
                                                   calculate pid();
                                                   battery voltage = battery voltage * 0.92 +
  pid last pitch d error = 0;
  pid i mem yaw = 0;
                                                  (analogRead(0) + 65) * 0.09853;
```

```
if(battery voltage < 1000 &&
                                                     esc 4 = 1000;
battery voltage > 600)digitalWrite(12,
                                                   if(micros() - loop timer >
                                                  4050)digitalWrite(12, HIGH);
HIGH);
 throttle = receiver input channel 3;
                                                    while(micros() - loop timer < 4000);
                                                    loop timer = micros();
 if (start == 2)
  if (throttle > 1800) throttle = 1800;
                                                    PORTD |= B11110000;
  esc 1 = throttle - pid output pitch +
                                                    timer channel 1 = \sec 1 + \text{loop timer};
                                                    timer channel 2 = \sec 2 + \text{loop timer};
pid output roll - pid output yaw;
  esc 2 = throttle + pid output pitch +
                                                    timer channel 3 = \sec 3 + \text{loop timer};
pid output roll + pid output yaw;
                                                    timer channel 4 = \sec 4 + \text{loop timer};
  esc 3 = \text{throttle} + \text{pid} output pitch -
                                                    gyro signalen();
pid output roll - pid output yaw;
  esc 4 = throttle - pid output pitch -
                                                    while(PORTD \geq 16)
pid output roll + pid output yaw;
                                                     esc loop timer = micros();
  if (battery voltage < 1240 &&
                                                     if(timer channel 1 <=
battery voltage > 800)
                                                  esc loop timer)PORTD &= B11101111;
   esc 1 += esc 1 * ((1240 -
                                                     if(timer channel 2 <=
battery voltage)/(float)3500);
                                                  esc loop timer)PORTD &= B11011111;
   esc 2 += esc 2 * ((1240 -
                                                     if(timer channel 3 <=
                                                  esc loop timer)PORTD &= B10111111;
battery voltage)/(float)3500);
   esc 3 += esc 3 * ((1240 -
                                                     if(timer channel 4 <=
battery voltage)/(float)3500);
                                                  esc loop timer)PORTD &= B01111111;
   esc 4 += esc 4 * ((1240 -
battery voltage)/(float)3500);
                                                  ISR(PCINT0 vect){
                                                   current time = micros();
  if (esc 1 < 1100) esc 1 = 1100;
  if (esc 2 < 1100) esc 2 = 1100;
                                                   //Channel
  if (esc 3 < 1100) esc 3 = 1100;
  if (esc 4 < 1100) esc 4 = 1100;
  if(esc 1 > 2000)esc 1 = 2000;
                                                   if(PINB & B00000001){
  if(esc 2 > 2000)esc 2 = 2000;
                                                     if(last channel 1 == 0){
                                                      last_channel_1 = 1;
  if(esc 3 > 2000)esc 3 = 2000;
  if(esc 4 > 2000)esc 4 = 2000;
                                                      timer 1 = current time;
 else{
                                                    else if(last channel 1 == 1){
  esc 1 = 1000;
                                                     last channel 1 = 0;
  esc 2 = 1000;
  esc 3 = 1000;
```

```
receiver input[1] = current time -
timer 1;
                                                   else if(last channel 4 == 1){
 }
                                                    last channel 4 = 0;
 //Channel
                                                    receiver input[4] = current time -
                                                  timer 4;
 if(PINB & B00000010){
  if(last channel 2 == 0){
   last_channel_2 = 1;
                                                  void gyro signalen(){
   timer 2 = current time;
                                                   //Read the MPU-6050
                                                   if(eeprom data[31] == 1)
                                                    Wire.beginTransmission(gyro address);
 else if(last channel 2 == 1){
                                                    Wire.write(0x3B);
  last channel 2 = 0;
                                                    Wire.endTransmission();
  receiver input[2] = current time -
                                                    Wire.requestFrom(gyro address, 14);
                                                    receiver input channel 1 =
timer 2;
                                                  convert receiver channel(1);
 }
 //Channel
                                                    receiver input channel 2 =
                                                  convert receiver channel(2);
                                                    receiver input channel 3 =
 if(PINB & B00000100){
                                                  convert receiver channel(3);
  if(last channel 3 == 0){
                                                    receiver input channel 4 =
   last channel 3 = 1;
                                                  convert receiver channel(4);
   timer 3 = \text{current time};
                                                    while(Wire.available() < 14);
                                                    acc axis[1] = Wire.read()<<8|Wire.read();
                                                    acc axis[2] = Wire.read()<<8|Wire.read();
                                                    acc axis[3] = Wire.read()<<8|Wire.read();</pre>
 else if(last channel 3 == 1){
  last channel 3 = 0;
                                                    temperature =
  receiver input[3] = current time -
                                                  Wire.read()<<8|Wire.read();
timer 3;
                                                    gyro axis[1] =
                                                  Wire.read()<<8|Wire.read();
 //Channel
                                                    gyro axis[2] =
                                                  Wire.read()<<8|Wire.read();
                                                    gyro axis[3] =
                                                  Wire.read()<<8|Wire.read();
 if(PINB & B00001000){
  if(last channel 4 == 0){
   last channel 4 = 1;
                                                   if(cal int == 2000)
   timer 4 = current time;
                                                    gyro axis[1] -= gyro axis cal[1];
```

```
gyro axis[2] -= gyro axis cal[2];
                                                  else if(pid i mem roll < pid max roll * -
                                                 1)pid i mem roll = pid max roll * -1;
  gyro axis[3] -= gyro axis cal[3];
                                                 pid output roll = pid p gain roll *
                                                pid error temp + pid i mem roll +
 gyro roll = gyro axis[eeprom data[28] &
                                                pid_d_gain_roll * (pid_error_temp -
0b00000011];
 if(eeprom data[28] &
                                                pid last roll d error);
0b10000000)gyro roll *= -1;
                                                  if(pid output roll >
                                                pid max roll)pid output roll =
 gyro pitch = gyro axis[eeprom data[29]]
& 0b00000011];
                                                pid max roll;
 if(eeprom data[29] &
                                                  else if(pid output roll < pid max roll * -
0b10000000)gyro pitch *= -1;
                                                 1)pid output roll = pid max roll * -1;
 gyro yaw = gyro axis[eeprom data[30] &
                                                 pid last roll d error = pid error temp;
0b00000011];
                                                 pid error temp = gyro pitch input -
 if(eeprom data[30] &
                                                pid pitch setpoint;
0b10000000)gyro yaw *= -1;
                                                  pid_i_mem_pitch += pid_i_gain_pitch *
 acc x = acc axis[eeprom data[29] &
                                                pid error temp;
0b00000011];
                                                  if(pid i mem pitch >
 if(eeprom_data[29] & 0b10000000)acc_x
                                                pid max pitch)pid i mem pitch =
*= -1:
                                                pid max pitch;
                                                  else if(pid i mem pitch < pid max pitch
 acc y = acc axis[eeprom data[28] &
                                                * -1)pid i mem pitch = pid max pitch * -
0b00000011];
 if(eeprom data[28] & 0b10000000)acc y
                                                 1;
*= -1;
 acc_z = acc_axis[eeprom_data[30] &
                                                 pid output pitch = pid p gain pitch *
0b00000011];
                                                pid error temp + pid i mem pitch +
 if(eeprom data[30] & 0b10000000)acc z
                                                pid_d_gain_pitch * (pid_error_temp -
*= -1;
                                                pid last pitch d error);
                                                  if(pid output pitch >
void calculate pid(){
                                                pid max pitch)pid output pitch =
 //Roll calculations
                                                pid max pitch;
                                                  else if(pid_output_pitch < pid_max_pitch *
 pid error temp = gyro roll input -
pid roll setpoint;
                                                -1)pid output pitch = pid max pitch * -1;
 pid_i mem_roll += pid_i gain_roll *
pid error temp;
                                                 pid last pitch d error = pid error temp;
 if(pid_i_mem_roll >
pid max roll)pid i mem roll =
                                                  pid error temp = gyro yaw input -
pid max roll;
                                                pid yaw setpoint;
```

```
pid_i_mem_yaw += pid_i_gain_yaw *
                                                    difference = ((long)(center - actual) *
                                                 (long)500) / (center - low);
pid error temp;
 if(pid i mem yaw >
                                                    if(reverse == 1)return 1500 + difference;
                                                    else return 1500 - difference;
pid max yaw)pid i mem yaw =
pid max yaw;
 else if(pid_i_mem_yaw < pid_max_yaw *
                                                  else if(actual > center){
-1)pid_i_mem_yaw = pid_max_yaw * -1;
                                                   if(actual > high)actual = high;
 pid output yaw = pid p gain yaw *
                                                    difference = ((long)(actual - center) *
pid error temp + pid i mem yaw +
                                                 (long)500) / (high - center);
pid_d gain_yaw * (pid_error_temp -
                                                    else return 1500 + difference;
pid last yaw d error);
 if(pid output yaw >
                                                  else return 1500;
pid max yaw)pid output yaw =
pid max yaw;
                                                 void set gyro registers(){
 else if(pid output yaw < pid max yaw * -
                                                  if(eeprom data[31] == 1){
1)pid output yaw = pid max yaw * -1;
                                                    Wire.beginTransmission(gyro address);
 pid last yaw d error = pid error temp;
                                                    Wire.write(0x6B);
                                                    Wire.write(0x00);
int convert receiver channel(byte
                                                    Wire.endTransmission();
function){
                                                    Wire.beginTransmission(gyro address);
 byte channel, reverse;
                                                    Wire.write(0x1B);
 int low, center, high, actual;
                                                    Wire.write(0x08);
                                                    Wire.endTransmission();
 int difference;
                                                    Wire.beginTransmission(gyro\_address);\\
 channel = eeprom_data[function + 23] &
0b00000111;
                                                    Wire.write(0x1C);
 if(eeprom_data[function + 23] &
                                                    Wire.write(0x10);
0b10000000)reverse = 1;
                                                    Wire.endTransmission();
                                                    Wire.beginTransmission(gyro address);
 else reverse = 0;
                                                    Wire.write(0x1B);
 actual = receiver input[channel];
 low = (eeprom data[channel * 2 + 15] <<
                                                    Wire.endTransmission();
8) | eeprom data[channel *2 + 14];
                                                    Wire.requestFrom(gyro address, 1);
 center = (eeprom data[channel * 2 - 1] <<
                                                    while(Wire.available() < 1);
8) | eeprom data[channel * 2 - 2];
                                                   if(Wire.read() != 0x08){
 high = (eeprom data[channel *2 + 7] <<
                                                     digitalWrite(12,HIGH);
8) | eeprom data[channel *2 + 6];
                                                     while(1)delay(10);
 if(actual < center){
  if(actual < low)actual = low;
                                                    Wire.beginTransmission(gyro\_address);\\
                                                    Wire.write(0x1A);
```

```
Wire.write(0x03);
                                                  void setup(){
  Wire.endTransmission();
                                                   Serial.begin(57600);
                                                   Wire.begin();
 }
                                                   TWBR = 12:
                                                   DDRD |= B11110000;
\wedge \wedge \wedge
                                                   DDRB = B00010000;
                                                   PCICR |= (1 << PCIE0);
#include <Wire.h>
#include <EEPROM.h>
                                                   PCMSK0 = (1 \leq PCINT0);
byte last channel 1, last channel 2,
                                                   PCMSK0 = (1 << PCINT1);
last channel 3, last channel 4;
                                                   PCMSK0 |= (1 << PCINT2);
byte eeprom data[36], start, data;
                                                   PCMSK0 |= (1 << PCINT3);
boolean new function request, first angle;
                                                   for(data = 0; data \leq 35;
volatile int receiver input channel 1,
                                                  data++)eeprom data[data] =
receiver input channel 2,
                                                  EEPROM.read(data);
receiver input channel 3,
                                                   gyro address = eeprom data[32];
receiver input channel 4;
                                                   set gyro registers();
int esc 1, esc 2, esc 3, esc 4;
                                                   //Check the EEPROM signature to make
int counter channel 1, counter channel 2,
                                                  sure that the setup program is executed.
counter channel 3, counter channel 4;
                                                   while(eeprom data[33]!='J'||
                                                  eeprom data[34] != 'M' || eeprom data[35]
int receiver input[5];
int loop counter, gyro address,
                                                  !='B'){}
vibration counter;
                                                    delay(500);
int temperature;
                                                     digitalWrite(12, !digitalRead(12));
long acc x, acc y, acc z,
acc total vector[20], acc av vector,
                                                   wait for receiver();
vibration total result;
                                                   zero timer = micros();
unsigned long timer channel 1,
timer channel 2, timer channel 3,
                                                   while(Serial.available())data =
                                                  Serial.read();
timer channel 4, esc timer,
esc loop timer;
                                                   data = 0;
unsigned long zero timer, timer 1, timer 2,
timer 3, timer 4, current time;
                                                  void loop(){
int acc axis[4], gyro axis[4];
                                                   while(zero timer +4000 > micros());
double gyro pitch, gyro roll, gyro yaw;
                                                   zero timer = micros();
                                                   if(Serial.available() > 0){
float angle roll acc, angle pitch acc,
angle pitch, angle roll;
                                                    data = Serial.read();
int cal int;
                                                    delay(100);
double gyro axis cal[4];
```

```
while (Serial.available () > 0) loop counter
                                                    if(receiver input channel 3 <
= Serial.read();
                                                   1025)new function request = false;
  new_function_request = true;
                                                    if(data == 0 && new function request ==
  loop counter = 0;
                                                   false){
  cal int = 0;
                                                     receiver input channel 3 =
                                                   convert receiver channel(3);
  start = 0;
  first angle = false;
                                                     esc 1 = receiver input channel 3;
  if(data == 'r')Serial.println("Reading
                                                     esc 2 = receiver input channel 3;
receiver signals.");
                                                     esc 3 = receiver input channel 3;
  if(data == 'a')Serial.println("Print the
                                                     esc 4 = receiver input channel 3;
quadcopter angles.");
                                                     esc pulse output();
  if(data == 'a')Serial.println("Gyro
calibration starts in 2 seconds (don't move
                                                    if(data == 'r')
the quadcopter).");
                                                     loop counter ++;
  if(data == '1')Serial.println("Test motor 1
                                                     receiver input channel 1 =
(right front CCW.)");
                                                   convert receiver channel(1);
  if(data == '2')Serial.println("Test motor 2
                                                     receiver input channel 2 =
                                                   convert receiver channel(2);
(right rear CW.)");
  if(data == '3')Serial.println("Test motor 3
                                                     receiver input channel 3 =
(left rear CCW.)");
                                                   convert receiver channel(3);
  if(data == '4')Serial.println("Test motor 4
                                                     receiver input channel 4 =
(left front CW.)");
                                                   convert receiver channel(4);
  if(data == '5')Serial.println("Test all
                                                     if(loop\ counter == 125){
                                                      print_signals();
motors together");
  for(vibration counter = 0;
                                                      loop counter = 0;
vibration_counter < 625;
vibration counter++){
                                                     if(receiver input channel 3 < 1050 &&
                                                   receiver input channel 4 < 1050) start = 1;
   delay(3);
                                                     if(start == 1 \&\&
   esc 1 = 1000;
   esc 2 = 1000;
                                                   receiver input channel 3 < 1050 &&
                                                   receiver input channel 4 > 1450) start = 2;
   esc 3 = 1000;
   esc 4 = 1000;
                                                     if(start == 2 \&\&
   esc pulse output();
                                                   receiver input channel 3 < 1050 &&
                                                   receiver input channel 4 > 1950) start = 0;
  vibration counter = 0;
                                                     esc 1 = 1000;
                                                     esc 2 = 1000;
 receiver input channel 3 =
                                                     esc 3 = 1000;
convert receiver channel(3);
                                                     esc 4 = 1000;
```

```
esc pulse output();
                                                         acc z = Wire.read() << 8 | Wire.read();
                                                         acc total vector[0] =
 if(data == '1' || data == '2' || data == '3' ||
                                                    sqrt((acc x*acc x)+(acc_y*acc_y)+(acc_z*
data == '4' || data == '5'){
                                                    acc z));
  loop counter ++;
                                                         acc av vector = acc total vector[0];
  if(new function request == true &&
                                                         for(start = 16; start > 0; start--){
loop counter == 250){
                                                          acc total vector[start] =
    Serial.print("Set throttle to 1000 (low).
                                                    acc total vector[start - 1];
It's now set to: ");
                                                          acc av vector +=
                                                    acc total vector[start];
Serial.println(receiver input channel 3);
    loop counter = 0;
                                                         acc av vector /= 17;
                                                         if(vibration counter < 20){
                                                          vibration counter ++;
  if(new function request == false){
                                                          vibration total result +=
   receiver input channel 3 =
convert receiver channel(3);
                                                    abs(acc total vector[0] - acc av vector);
    if(data == '1' || data == '5')esc 1 =
receiver input channel 3;
                                                         else{
   else esc 1 = 1000;
                                                          vibration counter = 0;
   if(data == '2' || data == '5')esc 2 =
receiver input channel 3;
                                                    Serial.println(vibration total result/50);
    else esc 2 = 1000;
                                                          vibration total result = 0;
   if(data == '3' || data == '5')esc_ 3 =
receiver input channel 3;
   else esc 3 = 1000;
   if(data == '4' || data == '5')esc 4 =
                                                     if(data == 'a'){
receiver input channel 3;
   else esc 4 = 1000;
   esc pulse output();
                                                      if(cal int != 2000)
   if(eeprom data[31] == 1){
                                                        Serial.print("Calibrating the gyro");
                                                        for (cal int = 0; cal int < 2000; cal int
Wire.beginTransmission(gyro address);
                                                    ++){
     Wire.write(0x3B);
                                                          digitalWrite(12, !digitalRead(12));
     Wire.endTransmission();
                                                    //Change the led status to indicate
     Wire.requestFrom(gyro address,6);
                                                    calibration.
     while(Wire.available() \leq 6);
                                                          Serial.print(".");
     acc x = Wire.read() << 8 | Wire.read();
     acc y = Wire.read()<<8|Wire.read();
                                                         gyro signalen();
```

```
gyro axis cal[1] += gyro axis[1];
                                                      angle roll acc =
                                                  asin((float)acc x/acc total vector[0])* -
    gyro axis cal[2] += gyro_axis[2];
    gyro axis cal[3] += gyro axis[3];
                                                  57.296;
    //We don't want the esc's to be beeping
                                                     if(!first angle){
annoyingly. So let's give them a 1000us puls
                                                       angle pitch = angle pitch acc;
                                                       angle roll = angle roll acc;
while calibrating the gyro.
    PORTD |= B11110000;
                                                       first angle = true;
    delayMicroseconds(1000);
    PORTD &= B00001111;
                                                      else{
                                                       angle pitch = angle pitch * 0.9996 +
    delay(3);
                                                  angle pitch acc * 0.0004;
                                                       angle roll = angle roll * 0.9996 +
   Serial.println(".");
   //Now that we have 2000 measures, we
                                                  angle roll acc * 0.0004;
need to devide by 2000 to get the average
gyro offset.
                                                      if(loop counter == 0)Serial.print("Pitch:
   gyro axis cal[1] /= 2000;
                                                  ");
   gyro axis cal[2] /= 2000;
                                                      if(loop counter ==
   gyro axis cal[3] /= 2000;
                                                  1)Serial.print(angle pitch ,0);
                                                     if(loop counter == 2)Serial.print(" Roll:
                                                  ");
  else{
                                                      if(loop counter ==
                                                  3)Serial.print(angle roll,0);
   PORTD |= B11110000;
                                                      if(loop counter == 4)Serial.print(" Yaw:
   delayMicroseconds(1000);
                                                  ");
   PORTD &= B00001111;
   gyro signalen();
                                                      if(loop counter ==
   angle pitch += gyro pitch * 0.0000611;
                                                  5)Serial.println(gyro yaw / 65.5,0);
   angle roll += gyro roll * 0.0000611;
                                                      loop counter ++;
                                                     if(loop counter == 60)loop counter = 0;
   angle pitch -= angle roll *
sin(gyro yaw * 0.000001066);
   angle roll += angle pitch *
sin(gyro yaw * 0.000001066);
                                                  ISR(PCINT0 vect){
   acc total vector[0] =
sqrt((acc x*acc x)+(acc y*acc y)+(acc z*
                                                   current time = micros();
acc z));
                                                   //Channel 1=====
                                                   if(PINB & B00000001){
   angle pitch acc =
                                                    if(last\_channel\_1 == 0){
asin((float)acc y/acc total vector[0])*
                                                     last channel_1 = 1;
57.296;
                                                     timer 1 = \text{current time};
```

```
else if(last channel 4 == 1){
  }
                                                    last channel 4 = 0;
 else if(last channel 1 == 1){
                                                    receiver input[4] = current time -
  last channel 1 = 0;
                                                  timer 4;
  receiver input[1] = current time -
timer 1;
                                                  void wait for receiver(){
 }
                                                   byte zero = 0;
 //Channel 2====
                                                   while(zero < 15){
 if(PINB & B00000010){
  if(last channel 2 == 0){
                                                    if(receiver input[1] < 2100 &&
   last channel 2 = 1;
                                                  receiver input[1] > 900)zero |=
   timer 2 = current time;
                                                  0b00000001;
                                                    if(receiver input[2] < 2100 \&\&
                                                  receiver input[2] > 900)zero |=
 else if(last channel 2 == 1){
                                                  0b0000010;
  last channel 2 = 0;
                                                    if(receiver input[3] < 2100 &&
  receiver input[2] = current time -
                                                  receiver input[3] > 900)zero |=
timer 2;
                                                  0b00000100;
 }
                                                    if(receiver input[4] < 2100 &&
                                                  receiver_input[4] > 900)zero |=
 //Channel 3=====
 if(PINB & B00000100){
                                                  0b00001000;
  if(last channel 3 == 0){
                                                    delay(500);
   last channel 3 = 1;
   timer 3 = current time;
                                                  int convert receiver channel(byte
                                                  function){
 else if(last channel 3 == 1){
                                                   byte channel, reverse;
  last channel 3 = 0;
                                                   int low, center, high, actual;
  receiver input[3] = current time -
                                                   int difference;
                                                   channel = eeprom data[function + 23] &
timer 3;
                                                  0b00000111;
 }
                                                   if(eeprom data[function + 23] &
 //Channel 4=====
 if(PINB & B00001000){
                                                  0b10000000)reverse = 1;
  if(last channel 4 == 0){
                                                   else reverse = 0;
   last channel 4 = 1;
                                                   actual = receiver input[channel];
   timer 4 = current time;
                                                   low = (eeprom data[channel * 2 + 15] <<
                                                  8) | eeprom data[channel *2 + 14];
```

```
center = (eeprom data[channel * 2 - 1] <<
                                                     if(receiver input channel 3 - 1480 <
8) | eeprom data[channel * 2 - 2];
                                                   0)Serial.print("vvv");
 high = (eeprom data[channel *2 + 7] <<
                                                     else if(receiver input channel 3 - 1520 >
8) | eeprom data[channel *2 + 6];
                                                   0)Serial.print("^^\");
                                                     else Serial.print("-+-");
 if(actual < center){
  if(actual < low)actual = low;
                                                     Serial.print(receiver input channel 3);
  difference = ((long)(center - actual) *
                                                     Serial.print(" Yaw:");
(long)500) / (center - low);
                                                     if(receiver input channel 4 - 1480 <
  if(reverse == 1)return 1500 + difference;
                                                   0)Serial.print("<<<");
  else return 1500 - difference;
                                                     else if(receiver input channel 4 - 1520 >
                                                   0)Serial.print(">>>");
 else if(actual > center){
                                                     else Serial.print("-+-");
  if(actual > high)actual = high;
                                                     Serial.println(receiver input channel 4);
  difference = ((long)(actual - center) *
(long)500) / (high - center);
                                                   void esc_pulse_output(){
  if(reverse == 1)return 1500 - difference;
                                                     zero timer = micros();
  else return 1500 + difference;
                                                     PORTD |= B11110000;
                                                     timer channel 1 = \sec 1 + \text{zero timer};
 else return 1500;
                                                     timer channel 2 = \sec 2 + \text{zero timer};
                                                     timer channel 3 = \sec 3 + \text{zero timer};
void print signals(){
                                                     timer channel 4 = \sec 4 + \text{zero timer};
 Serial.print("Start:");
                                                     while (PORTD \ge 16)
 Serial.print(start);
                                                      esc_loop_timer = micros();
                                                      if(timer channel 1 <=
 Serial.print(" Roll:");
 if(receiver input channel 1 - 1480 <
                                                   esc loop timer)PORTD &= B11101111;
0)Serial.print("<<<");
                                                      if(timer channel 2 <=
 else if(receiver input channel 1 - 1520 >
                                                   esc loop timer)PORTD &= B11011111;
0)Serial.print(">>>");
                                                      if(timer channel 3 <=
                                                   esc_loop_timer)PORTD &= B10111111;
 else Serial.print("-+-");
 Serial.print(receiver input channel 1);
                                                      if(timer channel 4 <=
 Serial.print(" Pitch:");
                                                   esc loop timer)PORTD &= B01111111;
 if(receiver input channel 2 - 1480 <
0)Serial.print("^^^");
                                                   void set gyro registers(){
 else if(receiver input channel 2 - 1520 >
                                                     if(eeprom data[31] == 1){
0)Serial.print("vvv");
                                                      Wire.beginTransmission(gyro address);
 else Serial.print("-+-");
                                                      Wire.write(0x6B);
                                                      Wire.write(0x00);
 Serial.print(receiver input channel 2);
 Serial.print(" Throttle:");
                                                      Wire.endTransmission();
```

```
Wire.beginTransmission(gyro_address);
                                                  gyro axis[2] =
                                               Wire.read()<<8|Wire.read();
  Wire.write(0x1B);
  Wire.write(0x08);
                                                  gyro axis[3] =
  Wire.endTransmission();
                                                Wire.read()<<8|Wire.read();
  Wire.beginTransmission(gyro address);
  Wire.write(0x1C);
                                                 if(cal int == 2000)
  Wire.write(0x10);
                                                  gyro axis[1] -= gyro axis cal[1];
                                                  gyro axis[2] -= gyro axis cal[2];
  Wire.endTransmission();
  Wire.beginTransmission(gyro address);
                                                  gyro axis[3] -= gyro axis cal[3];
  Wire.write(0x1B);
  Wire.endTransmission();
                                                 gyro roll = gyro axis[eeprom data[28] &
  Wire.requestFrom(gyro address, 1);
                                               0b00000011];
  while(Wire.available() \leq 1);
                                                 if(eeprom data[28] &
  if(Wire.read() != 0x08){
                                               0b10000000)gyro roll *= -1;
                                                gyro_pitch = gyro_axis[eeprom_data[29]
   digitalWrite(12,HIGH);
   while(1)delay(10);
                                               & 0b00000011];
                                                 if(eeprom data[29] &
                                               0b10000000)gyro_pitch *= -1;
  Wire.beginTransmission(gyro address);
  Wire.write(0x1A);
                                                 gyro yaw = gyro axis[eeprom data[30] &
  Wire.write(0x03);
                                               0b00000011];
  Wire.endTransmission();
                                                if(eeprom data[30] &
                                               0b10000000)gyro yaw *= -1;
                                                acc_x = acc_axis[eeprom_data[29] &
}
                                               0b000000111;
void gyro signalen(){
                                                if(eeprom data[29] & 0b10000000)acc x
if(eeprom data[31] == 1){
                                                *= -1:
  Wire.beginTransmission(gyro address);
                                                acc y = acc axis[eeprom data[28] &
  Wire.write(0x3B);
                                               0b00000011];
                                                if(eeprom_data[28] & 0b10000000)acc_y
  Wire.endTransmission();
  Wire.requestFrom(gyro address,14);
                                                *= -1:
  while(Wire.available() < 14);
                                                acc z = acc axis[eeprom data[30] &
  acc axis[1] = Wire.read()<<8|Wire.read();
                                               0b000000111;
  acc axis[2] = Wire.read()<<8|Wire.read();
                                                if(eeprom data[30] & 0b10000000)acc z
  acc axis[3] = Wire.read()<<8|Wire.read();
                                                *= -1:
  temperature =
Wire.read()<<8|Wire.read();
                                                ^^^^^^^
  gyro axis[1] =
Wire.read()<<8|Wire.read();
                                               #include <Wire.h>
```

```
#include <EEPROM.h>
byte last channel 1, last channel 2,
                                                  Serial.println(F(""));
last channel 3, last channel 4;
byte lowByte, highByte, type, gyro address,
                                                 Serial.println(F("======
error, clockspeed ok;
byte channel 1 assign, channel 2 assign,
                                                 ="));
channel 3 assign, channel 4 assign;
                                                  Serial.println(F("System check"));
byte roll axis, pitch axis, yaw axis;
byte receiver check byte, gyro check byte;
                                                 Serial.println(F("======
volatile int receiver input channel 1,
receiver input channel 2,
                                                 ="));
receiver input channel 3,
                                                  delay(1000);
receiver input channel 4;
                                                  Serial.println(F("Checking I2C clock
int center channel 1, center channel 2,
                                                 speed."));
center channel 3, center channel 4;
                                                  delay(1000);
int high channel 1, high channel 2,
                                                  TWBR = 12;
high channel 3, high channel 4;
                                                  #if F CPU == 16000000L
int low channel 1, low channel 2,
                                                   clockspeed ok = 1;
low channel 3, low channel 4;
                                                  #endif
int address, cal int;
                                                  if(TWBR == 12 && clockspeed ok){
unsigned long timer, timer 1, timer 2,
                                                   Serial.println(F("I2C clock speed is
timer 3, timer 4, current time;
                                                 correctly set to 400kHz."));
float gyro pitch, gyro roll, gyro yaw;
float gyro roll cal, gyro pitch cal,
                                                  else{
                                                   Serial.println(F("I2C clock speed is not
gyro yaw cal;
void setup(){
                                                 set to 400kHz. (ERROR 8)"));
 pinMode(12, OUTPUT);
                                                   error = 1;
 PCICR |= (1 << PCIE0);
 PCMSK0 |= (1 << PCINT0);
                                                  if(error == 0)
                                                   Serial.println(F(""));
 PCMSK0 |= (1 << PCINT1);
 PCMSK0 |= (1 << PCINT2);
 PCMSK0 |= (1 << PCINT3);
                                                 Serial.println(F("======
 Wire.begin();
 Serial.begin(57600);
                                                 ="));
                                                   Serial.println(F("Transmitter setup"));
 delay(250);
void loop(){
                                                 Serial.println(F("======
 //Show the YMFC-3D V2 intro
 intro();
```

```
Serial.println(receiver input channel 4);
="));
                                                       Serial.println(F(""));
  delay(1000);
                                                       Serial.println(F(""));
  Serial.print(F("Checking for valid
receiver signals."));
                                                      if(error == 0)
  wait for receiver();
                                                       Serial.println(F("Move the throttle stick to
  Serial.println(F(""));
                                                    full throttle and back to center"));
                                                       //Check for throttle movement
 if(error == 0)
                                                       check receiver inputs(1);
  delay(2000);
                                                       Serial.print(F("Throttle is connected to
  Serial.println(F("Place all sticks and
                                                    digital input "));
subtrims in the center position within 10
                                                       Serial.println((channel 3 assign &
seconds."));
                                                    0b00000111) + 7);
  for(int i = 9; i > 0; i--){
                                                       if(channel 3 assign &
   delay(1000);
                                                    0b10000000)Serial.println(F("Channel
    Serial.print(i);
                                                    inverted = yes"));
                                                       else Serial.println(F("Channel inverted =
    Serial.print(" ");
                                                    no"));
  Serial.println(" ");
                                                       wait sticks zero();
  //Store the central stick positions
  center channel 1 =
                                                       Serial.println(F(""));
receiver input channel 1;
                                                       Serial.println(F(""));
  center channel 2 =
                                                       Serial.println(F("Move the roll stick to
receiver input channel 2;
                                                    simulate left wing up and back to center"));
  center channel 3 =
                                                       //Check for throttle movement
receiver input channel 3;
                                                       check receiver inputs(2);
  center channel 4 =
                                                       Serial.print(F("Roll is connected to digital
receiver input channel 4;
                                                    input "));
  Serial.println(F(""));
                                                       Serial.println((channel 1 assign &
  Serial.println(F("Center positions
                                                    0b00000111) + 7);
                                                       if(channel 1 assign &
stored."));
  Serial.print(F("Digital input 08 = "));
                                                    0b10000000)Serial.println(F("Channel
  Serial.println(receiver input channel 1);
                                                    inverted = yes"));
                                                       else Serial.println(F("Channel inverted =
  Serial.print(F("Digital input 09 = "));
  Serial.println(receiver input channel 2);
                                                    no"));
  Serial.print(F("Digital input 10 = "));
                                                       wait sticks zero();
  Serial.println(receiver input channel 3);
  Serial.print(F("Digital input 11 = "));
                                                      if(error == 0){
```

```
Serial.println(F(""));
                                                       Serial.println(F("Gently move all the
                                                    sticks simultaneously to their extends"));
  Serial.println(F(""));
  Serial.println(F("Move the pitch stick to
                                                       Serial.println(F("When ready put the
simulate nose up and back to center"));
                                                    sticks back in their center positions"));
                                                       //Register the min and max values of the
  //Check for throttle movement
                                                    receiver channels
  check receiver inputs(3);
  Serial.print(F("Pitch is connected to
                                                       register min max();
digital input "));
                                                       Serial.println(F(""));
  Serial.println((channel 2 assign &
                                                       Serial.println(F(""));
0b00000111) + 7);
                                                       Serial.println(F("High, low and center
  if(channel 2 assign &
                                                    values found during setup"));
0b10000000)Serial.println(F("Channel
                                                       Serial.print(F("Digital input 08 values:"));
inverted = yes"));
                                                       Serial.print(low channel 1);
  else Serial.println(F("Channel inverted =
                                                       Serial.print(F(" - "));
no"));
                                                       Serial.print(center channel 1);
                                                       Serial.print(F(" - "));
  wait sticks zero();
                                                       Serial.println(high_channel 1);
 if(error == 0)
                                                       Serial.print(F("Digital input 09 values:"));
  Serial.println(F(""));
                                                       Serial.print(low channel 2);
  Serial.println(F(""));
                                                       Serial.print(F(" - "));
  Serial.println(F("Move the yaw stick to
                                                       Serial.print(center channel 2);
simulate nose right and back to center"));
                                                       Serial.print(F(" - "));
  //Check for throttle movement
                                                       Serial.println(high channel 2);
                                                       Serial.print(F("Digital input 10 values:"));
  check receiver inputs(4);
  Serial.print(F("Yaw is connected to digital
                                                       Serial.print(low channel 3);
input "));
                                                       Serial.print(F(" - "));
                                                       Serial.print(center channel 3);
  Serial.println((channel 4 assign &
0b00000111) + 7);
                                                       Serial.print(F(" - "));
                                                       Serial.println(high_channel 3);
  if(channel 4 assign &
0b10000000)Serial.println(F("Channel
                                                       Serial.print(F("Digital input 11 values:"));
inverted = yes"));
                                                       Serial.print(low channel 4);
  else Serial.println(F("Channel inverted =
                                                       Serial.print(F(" - "));
                                                       Serial.print(center channel 4);
no"));
  wait sticks zero();
                                                       Serial.print(F(" - "));
                                                       Serial.println(high channel 4);
                                                       Serial.println(F("Move stick 'nose up' and
 if(error == 0)
  Serial.println(F(""));
                                                    back to center to continue"));
  Serial.println(F(""));
                                                       check to continue();
```

```
}
                                                     Serial.println(F("Searching for
                                                 L3G4200D on address 0x68/104"));
 if(error == 0)
                                                     delay(1000);
  //What gyro is connected
                                                    if(search gyro(0x68, 0x0F) == 0xD3){
  Serial.println(F(""));
                                                      Serial.println(F("L3G4200D found on
                                                 address 0x68"));
Serial.println(F("===
                                                      type = 2;
                                                      gyro address = 0x68;
="));
  Serial.println(F("Gyro search"));
                                                   if(type == 0)
Serial.println(F("======
                                                     Serial.println(F("Searching for
                                                 L3G4200D on address 0x69/105");
="));
                                                     delay(1000);
  delay(2000);
                                                    if(search gyro(0x69, 0x0F) == 0xD3){
                                                      Serial.println(F("L3G4200D found on
  Serial.println(F("Searching for MPU-
                                                 address 0x69"));
6050 on address 0x68/104");
                                                      type = 2;
  delay(1000);
                                                      gyro address = 0x69;
  if(search gyro(0x68, 0x75) == 0x68){
                                                     }
   Serial.println(F("MPU-6050 found on
                                                   if(type == 0){
address 0x68"));
                                                     Serial.println(F("Searching for
   type = 1;
   gyro address = 0x68;
                                                 L3GD20H on address 0x6A/106");
                                                     delay(1000);
                                                    if(search gyro(0x6A, 0x0F) == 0xD7){
  if(type == 0)
                                                      Serial.println(F("L3GD20H found on
   Serial.println(F("Searching for MPU-
                                                 address 0x6A"));
6050 on address 0x69/105"));
                                                      type = 3;
   delay(1000);
                                                      gyro address = 0x6A;
   if(search gyro(0x69, 0x75) == 0x68){
    Serial.println(F("MPU-6050 found on
address 0x69"));
                                                   if(type == 0)
                                                    Serial.println(F("Searching for
    type = 1;
                                                 L3GD20H on address 0x6B/107"));
    gyro address = 0x69;
                                                    delay(1000);
                                                    if(search gyro(0x6B, 0x0F) == 0xD7){
  if(type == 0)
```

```
Serial.println(F("L3GD20H found on
                                                   ="));
address 0x6B"));
     type = 3;
                                                     Serial.println(F("Don't move the
     gyro address = 0x6B;
                                                   quadcopter!! Calibration starts in 3
                                                   seconds"));
                                                     delay(3000);
  if(type == 0)
                                                     Serial.println(F("Calibrating the gyro, this
                                                   will take +/- 8 seconds"));
   Serial.println(F("No gyro device
found!!! (ERROR 3)"));
                                                     Serial.print(F("Please wait"));
                                                     for (cal int = 0; cal int \leq 2000; cal int
   error = 1;
                                                   ++){
                                                       if(cal int % 100 ==
  else{
   delay(3000);
                                                   0)Serial.print(F("."));
   Serial.println(F(""));
                                                       gyro signalen();
                                                       gyro roll cal += gyro roll;
Serial.println(F("========
                                                       gyro pitch cal += gyro pitch;
                                                       gyro yaw cal += gyro yaw;
="));
                                                       delay(4);
   Serial.println(F("Gyro register
settings"));
                                                     //Now that we have 2000 measures, we
                                                   need to devide by 2000 to get the average
Serial.println(F("======
                                                   gyro offset.
                                                     gyro roll cal /= 2000;
="));
                                                     gyro pitch cal /= 2000;
                                                     gyro yaw cal /= 2000;
   start gyro(); //Setup the gyro for further
                                                     Serial.println(F(""));
use
                                                     Serial.print(F("Axis 1 offset="));
                                                     Serial.println(gyro roll cal);
                                                     Serial.print(F("Axis 2 offset="));
 if(error == 0)
  delay(3000);
                                                     Serial.println(gyro pitch cal);
  Serial.println(F(""));
                                                     Serial.print(F("Axis 3 offset="));
                                                     Serial.println(gyro yaw cal);
                                                     Serial.println(F(""));
Serial.println(F("======
="));
  Serial.println(F("Gyro calibration"));
                                                   Serial.println(F("======
Serial.println(F("=====
                                                   ="));
```

```
Serial.println(F("Gyro axes
                                                        Serial.println(F("OK!"));
configuration"));
                                                        Serial.print(F("Angle detection = "));
                                                        Serial.println(pitch axis &
                                                    0b00000011);
Serial.println(F("=====
                                                       if(pitch axis &
                                                    0b1000000)Serial.println(F("Axis inverted
="));
                                                    = yes"));
                                                        else Serial.println(F("Axis inverted =
  //Detect the left wing up movement
  Serial.println(F("Lift the left side of the
                                                   no"));
quadcopter to a 45 degree angle within 10
                                                        Serial.println(F("Put the quadcopter
seconds"));
                                                    back in its original position"));
  //Check axis movement
                                                        Serial.println(F("Move stick 'nose up'
  check gyro axes(1);
                                                    and back to center to continue"));
  if(error == 0)
                                                        check to continue();
    Serial.println(F("OK!"));
                                                        Serial.println(F(""));
    Serial.print(F("Angle detection = "));
                                                        Serial.println(F(""));
    Serial.println(roll axis & 0b00000011);
                                                        Serial.println(F("Rotate the nose of the
   if(roll_axis &
                                                    quadcopter 45 degree to the right within 10
0b10000000)Serial.println(F("Axis inverted
                                                    seconds"));
= yes"));
                                                       //Check axis movement
    else Serial.println(F("Axis inverted =
                                                       check gyro axes(3);
no"));
                                                      if(error == 0){
    Serial.println(F("Put the quadcopter
                                                        Serial.println(F("OK!"));
back in its original position"));
    Serial.println(F("Move stick 'nose up'
                                                        Serial.print(F("Angle detection = "));
and back to center to continue"));
                                                        Serial.println(yaw axis & 0b00000011);
                                                       if(yaw axis &
   check to continue();
                                                    0b10000000)Serial.println(F("Axis inverted
   //Detect the nose up movement
                                                    = yes"));
    Serial.println(F(""));
                                                        else Serial.println(F("Axis inverted =
    Serial.println(F(""));
                                                    no"));
                                                        Serial.println(F("Put the quadcopter
    Serial.println(F("Lift the nose of the
quadcopter to a 45 degree angle within 10
                                                    back in its original position"));
seconds"));
                                                        Serial.println(F("Move stick 'nose up'
                                                    and back to center to continue"));
   //Check axis movement
                                                        check to continue();
   check gyro axes(2);
  if(error == 0)
```

```
if(error == 0)
                                                delay(1000);
  Serial.println(F(""));
                                                if(gyro check byte == 0b00000111){
                                                 Serial.println(F("Gyro axes ok"));
Serial.println(F("===========
                                                else{
  Serial.println(F("LED test"));
                                                 Serial.println(F("Gyro exes verification
                                              failed!!! (ERROR 7)"));
Serial.println(F("=======
                                                 error = 1;
  digitalWrite(12, HIGH);
  Serial.println(F("The LED should now be
                                               if(error == 0)
lit"));
                                                //If all is good, store the information in
  Serial.println(F("Move stick 'nose up' and
                                              the EEPROM
back to center to continue"));
                                                Serial.println(F(""));
  check to continue();
                                              digitalWrite(12, LOW);
                                                Serial.println(F("Storing EEPROM
 Serial.println(F(""));
                                              information"));
 if(error == 0)
                                              Serial.println(F("==========
Serial.println(F("=======""));
                                                Serial.println(F("Writing EEPROM"));
                                                delay(1000);
  Serial.println(F("Final setup check"));
                                                Serial.println(F("Done!"));
                                                EEPROM.write(0, center channel 1 &
Serial.println(F("===========
                                              0b11111111);
                                                EEPROM.write(1, center channel 1 >>
  delay(1000);
                                              8);
  if(receiver check byte == 0b00001111){
                                                EEPROM.write(2, center channel 2 &
   Serial.println(F("Receiver channels
                                              0b11111111);
ok"));
                                                EEPROM.write(3, center channel 2 >>
  }
                                              8);
                                                EEPROM.write(4, center channel 3 &
  else{
   Serial.println(F("Receiver channel
                                              0b11111111);
verification failed!!! (ERROR 6)"));
                                                EEPROM.write(5, center channel 3 >>
   error = 1;
                                              8);
  }
```

```
EEPROM.write(6, center channel 4 &
                                                EEPROM.write(27, channel 4 assign);
                                                EEPROM.write(28, roll axis);
0b11111111);
  EEPROM.write(7, center channel 4>>
                                                EEPROM.write(29, pitch axis);
                                                EEPROM.write(30, yaw axis);
8);
  EEPROM.write(8, high channel 1 &
                                                EEPROM.write(31, type);
                                                EEPROM.write(32, gyro address);
0b11111111);
  EEPROM.write(9, high channel 1 >> 8);
                                                //Write the EEPROM signature
  EEPROM.write(10, high channel 2 &
                                                EEPROM.write(33, 'J');
                                                EEPROM.write(34, 'M');
0b11111111);
  EEPROM.write(11, high channel 2 >>
                                                EEPROM.write(35, 'B');
                                                Serial.println(F("Verify EEPROM data"));
8);
  EEPROM.write(12, high channel 3 &
                                                delay(1000);
0b11111111);
                                                if(center channel 1 !=
  EEPROM.write(13, high_channel_3 >>
                                              ((EEPROM.read(1) << 8)
8);
                                              EEPROM.read(0))error = 1;
  EEPROM.write(14, high channel 4 &
                                                if(center channel 2 !=
0b11111111);
                                              ((EEPROM.read(3) << 8) \mid
  EEPROM.write(15, high channel 4>>
                                              EEPROM.read(2))error = 1;
8);
                                                if(center channel 3 !=
                                              ((EEPROM.read(5) << 8) |
  EEPROM.write(16, low channel 1 &
0b11111111);
                                              EEPROM.read(4)))error = 1;
  EEPROM.write(17, low channel 1 >>
                                                if(center channel 4 !=
                                              ((EEPROM.read(7) << 8)
8);
  EEPROM.write(18, low channel 2 &
                                              EEPROM.read(6)))error = 1;
0b11111111);
  EEPROM.write(19, low channel 2 >>
                                                if(high channel 1 != ((EEPROM.read(9)
                                              << 8) | EEPROM.read(8)))error = 1;
8);
  EEPROM.write(20, low channel 3 &
                                                if(high channel 2!=
0b11111111);
                                              ((EEPROM.read(11) << 8) \mid
  EEPROM.write(21, low channel 3 >>
                                              EEPROM.read(10)))error = 1;
                                                if(high channel 3!=
8);
  EEPROM.write(22, low channel 4 &
                                              ((EEPROM.read(13) << 8) |
                                              EEPROM.read(12)))error = 1;
0b11111111);
  EEPROM.write(23, low channel 4>>
                                                if(high channel 4!=
                                              ((EEPROM.read(15) << 8) |
8);
  EEPROM.write(24, channel 1 assign);
                                              EEPROM.read(14))error = 1;
  EEPROM.write(25, channel 2 assign);
  EEPROM.write(26, channel 3 assign);
```

```
if(low channel 1 != ((EEPROM.read(17)
                                                    Serial.println(F("You can now calibrate
<< 8) | EEPROM.read(16)))error = 1;
                                                 the esc's and upload the YMFC-AL code."));
  if(low channel 2 != ((EEPROM.read(19)
<< 8) | EEPROM.read(18)))error = 1;
                                                   else{
  if(low channel 3 != ((EEPROM.read(21)
                                                   Serial.println(F("The setup is aborted due
<< 8) | EEPROM.read(20)))error = 1;
                                                 to an error."));
  if(low channel 4!=((EEPROM.read(23)
                                                   Serial.println(F("Check the Q and A page
<< 8) | EEPROM.read(22)))error = 1;
                                                 of the YMFC-AL project on:"));
                                                   Serial.println(F("www.brokking.net for
                                                 more information about this error."));
  if(channel_1_assign !=
EEPROM.read(24))error = 1;
  if(channel 2 assign!=
                                                   while(1);
EEPROM.read(25))error = 1;
  if(channel 3 assign!=
                                                 byte search_gyro(int gyro_address, int
EEPROM.read(26))error = 1;
                                                 who am i){
  if(channel 4 assign!=
                                                   Wire.beginTransmission(gyro_address);
EEPROM.read(27))error = 1;
                                                   Wire.write(who am i);
                                                   Wire.endTransmission();
  if(roll axis != EEPROM.read(28))error =
                                                   Wire.requestFrom(gyro address, 1);
                                                   timer = millis() + 100;
1;
  if(pitch axis != EEPROM.read(29))error
                                                   while(Wire.available() < 1 && timer >
                                                 millis());
  if(yaw_axis != EEPROM.read(30))error =
                                                   lowByte = Wire.read();
1;
                                                   address = gyro address;
  if(type != EEPROM.read(31))error = 1;
                                                   return lowByte;
  if(gyro address!=
EEPROM.read(32))error = 1;
                                                 void start gyro(){
                                                   if(type == 2 \parallel \text{type} == 3){
  if('J' != EEPROM.read(33))error = 1;
                                                    Wire.beginTransmission(address);
  if('M' != EEPROM.read(34))error = 1;
  if('B' != EEPROM.read(35))error = 1;
                                                    Wire.write(0x20);
                                                    Wire.write(0x0F);
  if(error == 1)Serial.println(F("EEPROM
                                                    Wire.endTransmission();
                                                    Wire.beginTransmission(address);
verification failed!!! (ERROR 5)"));
  else Serial.println(F("Verification done"));
                                                    Wire.write(0x20);
                                                    Wire.endTransmission();
 if(error == 0)
                                                    Wire.requestFrom(address, 1);
  Serial.println(F("Setup is finished."));
                                                    while(Wire.available() \leq 1);
```

```
Serial.print(F("Register 0x20 is set to:"));
 Serial.println(Wire.read(),BIN);
 Wire.beginTransmission(address);
                                                 void gyro signalen(){
                                                   if(type == 2 || type == 3){
 Wire.write(0x23);
 Wire.write(0x90);
                                                    Wire.beginTransmission(address);
 Wire.endTransmission();
                                                    Wire.write(168);
 Wire.beginTransmission(address);
                                                    Wire.endTransmission();
 Wire.write(0x23);
                                                    Wire.requestFrom(address, 6);
 Wire.endTransmission();
                                                    while (Wire.available () \leq 6);
 Wire.requestFrom(address, 1);
                                                    lowByte = Wire.read();
 while(Wire.available() \leq 1);
                                                    highByte = Wire.read();
 Serial.print(F("Register 0x23 is set to:"));
                                                    gyro roll = ((highByte << 8)|lowByte);
 Serial.println(Wire.read(),BIN);
                                                    if(cal int == 2000)gyro roll -=
                                                 gyro roll cal;
if(type == 1){
                                                    lowByte = Wire.read();
 Wire.beginTransmission(address);
                                                    highByte = Wire.read();
                                                    gyro pitch = ((highByte << 8)|lowByte);
 Wire.write(0x6B);
 Wire.write(0x00);
                                                    if(cal int == 2000)gyro pitch -=
 Wire.endTransmission();
                                                 gyro pitch cal;
 Wire.beginTransmission(address);
                                                    lowByte = Wire.read();
 Wire.write(0x6B);
                                                    highByte = Wire.read();
 Wire.endTransmission();
                                                    gyro yaw = ((highByte << 8)|lowByte);
 Wire.requestFrom(address, 1);
                                                    if(cal int == 2000)gyro yaw -=
 while(Wire.available() < 1);
                                                 gyro yaw cal;
 Serial.print(F("Register 0x6B is set to:"));
 Serial.println(Wire.read(),BIN);
                                                   if(type == 1)
 Wire.beginTransmission(address);
                                                    Wire.beginTransmission(address);
 Wire.write(0x1B);
                                                    Wire.write(0x43);
 Wire.write(0x08);
                                                    Wire.endTransmission();
 Wire.endTransmission();
                                                    Wire.requestFrom(address,6);
 Wire.beginTransmission(address);
                                                    while(Wire.available() < 6);
                                                    gyro roll=Wire.read()<<8|Wire.read();
 Wire.write(0x1B);
 Wire.endTransmission();
                                                    if(cal int == 2000)gyro roll ==
 Wire.requestFrom(address, 1);
                                                 gyro roll cal;
 while(Wire.available() \leq 1);
                                                    gyro pitch=Wire.read()<<8|Wire.read();
 Serial.print(F("Register 0x1B is set to:"));
                                                    if(cal int == 2000)gyro pitch -=
 Serial.println(Wire.read(),BIN);
                                                 gyro pitch cal;
```

```
gyro yaw=Wire.read()<<8|Wire.read();
                                                     pulse length =
  if(cal int == 2000)gyro yaw ==
                                                 receiver input channel 4;
gyro yaw cal;
                                                   if(trigger == 0){
                                                    error = 1;
void check_receiver_inputs(byte
                                                    Serial.println(F("No stick movement
movement){
                                                 detected in the last 30 seconds!!! (ERROR
 byte trigger = 0;
 int pulse length;
                                                 2)"));
 timer = millis() + 30000;
 while(timer > millis() && trigger == 0){
                                                  //Assign the stick to the function.
  delay(250);
                                                   else{
  if(receiver input channel 1 > 1750 ||
                                                    if(movement == 1){
                                                     channel_3_assign = trigger;
receiver_input_channel_1 < 1250){
   trigger = 1;
                                                     if(pulse length <
   receiver_check_byte |= 0b00000001;
                                                  1250)channel 3 assign += 0b10000000;
   pulse length =
receiver input channel 1;
                                                    if(movement == 2)
                                                     channel 1 assign = trigger;
                                                     if(pulse length <
  if(receiver input channel 2 > 1750 ||
receiver input channel 2 < 1250)
                                                  1250)channel 1 assign += 0b10000000;
   trigger = 2;
   receiver check byte = 0b00000010;
                                                    if(movement == 3)
                                                     channel_2_assign = trigger;
   pulse length =
receiver input channel 2;
                                                     if(pulse length <
                                                  1250)channel_2_assign += 0b10000000;
  if(receiver input channel 3 > 1750 ||
receiver input channel 3 < 1250)
                                                    if(movement == 4)
   trigger = 3;
                                                     channel 4 assign = trigger;
   receiver_check_byte |= 0b00000100;
                                                     if(pulse length <
                                                  1250)channel_4_assign += 0b10000000;
   pulse length =
receiver_input_channel_3;
  if(receiver_input_channel_4 > 1750 \parallel
receiver_input_channel_4 < 1250){
   trigger = 4;
                                                 void check_to_continue(){
   receiver_check_byte |= 0b00001000;
                                                  byte continue byte = 0;
                                                   while(continue byte == 0){
```

```
if(channel 2 assign == 0b00000001 \&\&
                                                   if(receiver input channel 2 <
receiver input channel 1>
                                                center channel 2 + 20 &&
center channel 1 + 150)continue byte = 1;
                                                receiver input channel 2>
  if(channel 2 assign == 0b10000001 &&
                                                center channel 2 - 20)zero |= 0b00000010;
receiver input channel 1 <
                                                   if(receiver input channel 3 <
center channel 1 - 150)continue byte = 1;
                                                center channel 3 + 20 &&
  if(channel 2 assign == 0b00000010 &&
                                                receiver input channel 3 >
receiver input channel 2>
                                                center channel 3 - 20)zero |= 0b00000100;
center channel 2 + 150)continue byte = 1;
                                                   if(receiver input channel 4 <
  if(channel 2 assign == 0b10000010 &&
                                                center channel 4 + 20 &&
receiver input channel 2 <
                                                receiver input channel 4>
center channel 2 - 150)continue byte = 1;
                                                center channel 4 - 20)zero |= 0b00001000;
  if(channel 2 assign == 0b00000011 &&
                                                   delay(100);
receiver input channel 3 >
center channel 3 + 150)continue byte = 1;
  if(channel_2_assign == 0b10000011 &&
receiver input channel 3 <
                                                //Checck if the receiver values are valid
center_channel_3 - 150)continue_byte = 1;
                                                within 10 seconds
  if(channel 2 assign == 0b00000100 \&\&
                                                void wait for receiver(){
                                                 byte zero = 0;
receiver input channel 4>
center channel 4 + 150)continue byte = 1;
                                                  timer = millis() + 10000;
  if(channel 2 assign == 0b10000100 &&
                                                  while(timer > millis() && zero < 15){
receiver input channel 4 <
                                                   if(receiver input channel 1 < 2100 &&
center channel 4 - 150)continue byte = 1;
                                                receiver input channel 1 > 900)zero |=
  delay(100);
                                                0b0000001;
                                                   if(receiver input channel 2 < 2100 &&
                                                receiver input channel 2 > 900)zero |=
 wait sticks zero();
                                                0b0000010;
                                                   if(receiver input channel 3 < 2100 &&
                                                receiver input channel 3 > 900)zero |=
//Check if the transmitter sticks are in the
neutral position
                                                0b00000100;
void wait sticks zero(){
                                                   if(receiver input channel 4 < 2100 &&
                                                receiver input channel 4 > 900)zero |=
 byte zero = 0;
 while (zero < 15)
                                                0b00001000;
  if(receiver input channel 1 <
                                                   delay(500);
center channel 1 + 20 &&
                                                   Serial.print(F("."));
receiver input channel 1>
center_channel_1 - 20)zero |= 0b00000001;
                                                 if(zero == 0)
```

```
receiver input channel 3 >
  error = 1;
  Serial.println(F("."));
                                                 center channel 3 - 20)zero |= 0b00000100;
  Serial.println(F("No valid receiver signals
                                                   if(receiver input channel 4 <
found!!! (ERROR 1)"));
                                                 center channel 4 + 20 &&
                                                 receiver input channel 4>
                                                 center channel 4 - 20)zero |= 0b00001000;
 else Serial.println(F(" OK"));
                                                   if(receiver input channel 1 <
                                                 low channel 1)low channel 1 =
//Register the min and max receiver values
                                                 receiver input channel 1;
and exit when the sticks are back in the
                                                   if(receiver input channel 2 <
neutral position
                                                 low channel 2)low channel 2 =
void register min max(){
                                                 receiver input channel 2;
 byte zero = 0;
                                                   if(receiver input channel 3 <
 low channel 1 =
                                                 low channel 3)low channel 3 =
receiver input channel 1;
                                                 receiver input channel 3;
 low channel 2 =
                                                   if(receiver input channel 4 <
                                                 low channel 4)low channel 4=
receiver input channel 2;
 low channel 3 =
                                                 receiver input channel 4;
receiver input channel 3;
                                                   if(receiver input channel 1 >
 low_channel 4 =
                                                 high channel 1)high channel 1 =
receiver input channel 4;
                                                receiver input channel 1;
 while(receiver input channel 1 <
                                                   if(receiver input channel 2 >
                                                high channel 2)high channel 2 =
center channel 1 + 20 &&
receiver_input_channel_1 >
                                                 receiver input channel 2;
center channel 1 - 20)delay(250);
                                                   if(receiver input channel 3 >
 Serial.println(F("Measuring
                                                high channel 3)high channel 3 =
endpoints...."));
                                                 receiver input channel 3;
                                                   if(receiver input channel 4>
 while(zero < 15){
                                                 high channel 4)high channel 4=
  if(receiver input channel 1 <
center channel 1 + 20 \&\&
                                                 receiver input channel 4;
receiver input channel 1>
                                                   delay(100);
center channel 1 - 20)zero |= 0b00000001;
                                                  }
  if(receiver input channel 2 <
                                                 s
center channel 2 + 20 &&
                                                 void check gyro axes(byte movement){
receiver input channel 2>
                                                  byte trigger axis = 0;
center channel 2 - 20)zero |= 0b00000010;
                                                  float gyro angle roll, gyro angle pitch,
                                                 gyro angle yaw;
  if(receiver input channel 3 <
center channel 3 + 20 &&
                                                  //Reset all axes
```

```
gyro angle roll = 0;
 gyro angle pitch = 0;
                                                  if((gyro angle pitch < -30 ||
 gyro_angle_yaw = 0;
                                                 gyro angle pitch > 30) && gyro angle roll
                                                 > -30 && gyro angle roll < 30 &&
 gyro signalen();
 timer = millis() + 10000;
                                                 gyro angle yaw > -30 && gyro angle yaw
 while(timer > millis() && gyro angle roll
                                                 < 30){
> -30 && gyro angle roll < 30 &&
                                                   gyro check byte |= 0b00000010;
gyro angle pitch > -30 &&
                                                   if(gyro angle pitch < 0)trigger axis =
gyro angle pitch < 30 && gyro angle yaw
                                                 0b10000010;
> -30 \&\& gyro angle yaw < 30){
                                                   else trigger axis = 0b00000010;
  gyro signalen();
  if(type == 2 || type == 3){
                                                  if((gyro angle yaw < -30 \parallel
   gyro angle roll += gyro roll * 0.00007;
                                                 gyro angle yaw > 30) && gyro angle roll
   gyro angle pitch += gyro pitch *
                                                 > -30 \&\& gyro angle roll < 30 \&\&
0.00007;
                                                 gyro angle pitch > -30 &&
                                                 gyro angle pitch < 30){
   gyro angle yaw += gyro yaw *
0.00007;
                                                   gyro check byte |= 0b00000100;
                                                   if(gyro angle yaw < 0)trigger axis =
  if(type == 1)
                                                 0b10000011;
   gyro angle roll += gyro roll *
                                                   else trigger axis = 0b00000011;
0.0000611;
   gyro_angle_pitch += gyro_pitch *
                                                  if(trigger\ axis == 0)
                                                   error = 1;
0.0000611;
   gyro_angle_yaw += gyro_yaw *
                                                   Serial.println(F("No angular motion is
                                                 detected in the last 10 seconds!!! (ERROR
0.0000611;
                                                 4)"));
  delayMicroseconds(3700);
                                                  else
 //Assign the moved axis to the
                                                  if(movement == 1)roll axis = trigger axis;
                                                  if(movement == 2)pitch axis =
orresponding function (pitch, roll, yaw)
 if((gyro angle roll \leq -30 || gyro angle roll
                                                 trigger axis;
> 30) && gyro angle pitch > -30 &&
                                                  if(movement == 3)yaw axis = trigger axis
gyro angle pitch < 30 && gyro angle yaw
> -30 \&\& gyro angle yaw < 30){
  gyro check byte = 0b00000001;
                                                 //This routine is called every time input 8, 9,
  if(gyro angle roll < 0)trigger axis =
                                                 10 or 11 changed state
                                                 ISR(PCINT0 vect){
0b10000001;
                                                  current time = micros();
  else trigger axis = 0b00000001;
```

```
last channel 4 = 1;
 //Channel 1======
 if(PINB & B00000001){
                                                     timer 4 = current time;
  if(last channel 1 == 0){
   last channel 1 = 1;
   timer 1 = current time;
                                                   else if(last channel 4 == 1){
                                                    last channel 4 = 0;
                                                    receiver input channel 4 = current time
 else if(last channel 1 == 1){
                                                  - timer 4;
  last channel 1 = 0;
  receiver input channel 1 = current time
- timer 1;
                                                  //Intro subroutine
                                                  void intro(){
 }
                                                  Serial.println(F("=====
 //Channel 2======
 if(PINB & B00000010){
  if(last channel 2 == 0){
                                                   delay(1500);
   last channel 2 = 1;
                                                   Serial.println(F(""));
   timer 2 = current time;
                                                   Serial.println(F("Your"));
                                                   delay(500);
                                                   Serial.println(F(" Multicopter"));
 else if(last channel 2 == 1){
                                                   delay(500);
  last channel 2 = 0;
                                                   Serial.println(F(" Flight"));
  receiver input channel 2 = current time
                                                   delay(500);
                                                   Serial.println(F("
- timer 2;
                                                                       Controller"));
 //Channel 3====
                                                   delay(1000);
 if(PINB & B00000100){
                                                   Serial.println(F(""));
  if(last channel 3 == 0){
                                                   Serial.println(F("YMFC-AL Setup
   last channel 3 = 1;
                                                  Program"));
   timer 3 = current time;
                                                   Serial.println(F(""));
                                                  Serial.println(F("======
                                                                                 ="));
 else if(last channel 3 == 1){
                                                   delay(1500);
  last channel 3 = 0;
                                                   Serial.println(F("For support and
  receiver input channel 3 = current time
                                                  questions: www.brokking.net"));
- timer 3;
                                                   Serial.println(F(""));
                                                   Serial.println(F("Have fun!"));
 //Channel 4==
 if(PINB & B00001000){
  if(last channel 4 == 0){
```