```
1 #include <Wire.h> //Include the Wire.h library so we can communicate with the gyro.
  2 #include <EEPROM.h>
     //#include <SoftwareSerial.h>
 4 #include <string.h>
5 #include "I2Cdev.h"
 6 #include "MPU6050.h'
  8 #define IBUS_BUFFSIZE 32 // Max iBus packet size (2 byte header, 14 channels x 2 bytes, 2 byte checksum)
 9 #define IBUS_MAXCHANNELS 6
10 static uint8_t ibusIndex = 0;
static uint8_t ibus[IBUS_BUFFSIZE] = {0};
12 static uint16_t rcValue[IBUS_MAXCHANNELS];
13 boolean rxFrameDone = false:
14 //SoftwareSerial mySerial(8, 12);
uint8_t p_light = 5;
16 uint8_t d_light = 1;
17 uint8_t i_light = 1;
uint16_t p_light_k, d_light_k, i_light_k, button_pushed;
19 int p_mul = 40, i_mul = 25, d_mul = 25;
21 int valueThrottle, valuePitch, valueRoll, valueYaw, valueAux1, valueAux2;
22
o uint8_t pid_p_gain_roll = 0; //Gain setting for the roll P-controller (1.3)
uint8_t pid_i_gain_roll = 0; //Gain setting for the roll I-controller (0.3)
uint8_t pid_d_gain_roll = 0; //Gain setting for the roll D-controller (15)
 29 // int pid_max_roll = 400;
                                        //Maximum output of the PID-controller (+/-)
 30
// float pid_p_gain_pitch = pid_p_gain_roll; //Gain setting for the pitch P-controller.
// float pid_i_gain_pitch = pid_i_gain_roll; //Gain setting for the pitch I-controller.
// float pid_d_gain_pitch = pid_d_gain_roll; //Gain setting for the pitch D-controller.
// int pid_max_pitch = pid_max_roll; //Maximum output of the PID-controller (+/-)
uint8_t pid_p_gain_yaw = 0; //Gain setting for the pitch P-controller. //4.0 uint8_t pid_i_gain_yaw = 0; //Gain setting for the pitch I-controller. //0.02 uint8_t pid_d_gain_yaw = 0; //Gain setting for the pitch D-controller.
39
    // int pid_max_yaw = 400;
                                       //Maximum output of the PID-controller (+/-)
40
intio_t telm = 0, itelm = 0, itelm; delta, axis,
sintio_t gyroInput[2], pidConfig[2], lastgyro[2], delta1[2], delta2[2], rc_command[2], pidOutput[2];
de // byte last_channel_1, last_channel_2, last_channel_3, last_channel_4;
// int counter_channel_1, counter_channel_2, counter_channel_3, counter_channel_4, loop_counter;
dint esc_1 = 1000, esc_2 = 1000, esc_3 = 1000, esc_4 = 1000;
49 int throttle, battery_voltage;
50 unsigned long timer_channel_1, timer_channel_2, timer_channel_3, timer_channel_4, esc_timer, esc_loop_timer;
51 unsigned long timer_1, timer_2, timer_3, timer_4, current_time;
52 int cal_int, start;
 53 unsigned long loop_timer;
54 int16_t gyro_pitch, gyro_roll, gyro_yaw, acc_pitch, acc_roll, acc_yaw, temperature;
    //double gyro_roll_cal , gyro_pitch_cal, gyro_yaw_cal;
int32_t gyro_roll_cal = 0, gyro_pitch_cal = 0, gyro_yaw_cal = 0;
55
57 byte highByte, lowByte;
58
 59 int16 t pid error temp:
60
61 int16_t pid_i_error[2], pid_roll_command, gyro_roll_meter, pid_output_roll, pid_last_roll_d_error;
62 int16_t pid_pitch_command, gyro_pitch_meter, pid_output_pitch, pid_last_pitch_d_error;
63 int16_t pid_i_mem_yaw, rc_yaw_command, gyro_yaw_meter, pid_output_yaw, pid_last_yaw_d_error;
64
65 #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
66 #include "Wire.h'
67 #endif
69 // class default I2C address is 0x68
70 // specific I2C addresses may be passed as a parameter here
71 // AD0 low = 0x68 (default for InvenSense evaluation board)
72 // AD0 high = 0x69
73 MPU6050 accelgyro;
75 int16_t ax, ay, az;
76 int16_t gx, gy, gz;
77 boolean readingRx = false;
78
80 //Setup routine
82 void setup()
83 {
      // mySerial.begin(115200);
Serial.begin(115200);
// join IZC bus (IZCdev library doesn't do this automatically)
84
85
87 #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
      Wire.begin();
88
       Wire.setClock(400000);
90 #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
91
      Fastwire::setup(400, true):
      accelgyro.initialize();
accelgyro.setDLPFMode(MPU6050_DLPF_BW_10);
93
94
       // verify connection
       pid_p_gain_roll = EEPROM.read(3);
pid_d_gain_roll = EEPROM.read(4);
96
97
98
       pid_i_gain_roll = EEPROM.read(5);
       pid_p_gain_yaw = EEPROM.read(6);
pid_d_gain_yaw = EEPROM.read(7);
99
100
101
       pid_i_gain_yaw = EEPROM.read(8);
102
       Wire.begin(); //Start the I2C as master.
       TWBR = 12:
103
```

```
105
      DDRD |= B11101000; //Configure digital port D pin 6 7 as output.
      DDRB |= B00101110; // pin 9 10 11 12 13 as output
106
107
      PORTC |= B00000111: // pull the 14 15 16 hight
108
109
      //Use the led on the Arduino for startup indication
110
      111
112
113
      114
115
116
117
           accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
gyro_roll_cal += gy;
                                                                                                   //Read the gyro output.
                                                             //Ad roll value to gyro_roll_cal.
118
      //
//
//
           119
120
121
      //
//
//
122
            PORTD |= B10000000;
                                                                   //Set digital poort 4, 5, 6 and 7 high.
123
           delayMicroseconds(1000):
                                                                   //Wait 1000us.
            PORTD &= B00000000;
                                                                   //Set digital poort 4, 5, 6 and 7 low.
124
      //
//
//
125
           delay(3);
                                                                   //Wait 3 milliseconds before the next loop.
         3
126
127
      ///
gyro_roll_cal = gyro_roll_cal / 2000;
// gyro_pitch_cal = gyro_pitch_cal / 2000;
// gyro_yaw_cal = gyro_yaw_cal / 2000;
128
                                                                         //Ad roll value to gyro_roll_cal.
129
                                                                         //Ad pitch value to gyro_pitch_cal.
130
131
      //
             Serial.print("a/g:\t");
132
             Serial.print(gyro_roll_cal); Serial.print("\t");
133
134
             Serial.print(gyro_pitch_cal); Serial.print("\t");
      11
135
             Serial.println(gyro_yaw_cal);
136 }
137
138
139
140
141
143
144
148
    void loop()
149 {
      loop timer = micros();
150
      if (readingRx == false) {
151
        readingRx == true;
while (micros() - loop_timer < 650) {</pre>
152
153
         if (Serial.available()) {
    uint8_t val = Serial.read(); //https://arduino.stackexchange.com/questions/10088/what-is-the-difference-between-serial-write-and-serial-
print-and-when-are-they/48147#48147?newreg=054f10a9de0f4013bb1fb4ed2115e030
155
           uint16_t chksum, rxsum;
157
           uint8_t i;
158
            // Look for 0x2040 as start of packet
160
            if (ibusIndex == 0 && val != 0x20) {
161
             return;
162
           if (ibusIndex == 1 && val != 0x40) {
  ibusIndex = 0;
163
164
                                                         // index to 0000
165
             return;
166
           3
167
168
           if (ibusIndex < IBUS_BUFFSIZE) {</pre>
169
             ibus[ibusIndex] = val;
                                                        // index increment
170
             ibusIndex++:
172
           3
173
           if (ibusIndex == IBUS BUFFSIZE) {
175
             ibusIndex = 0:
176
             chksum = 0xFFFF;
178
             for (i = 0; i < 30; i++) {
  chksum -= ibus[i];</pre>
179
             }
181
182
             rxsum = ibus[30] + (ibus[31] << 8);
184
             if (chksum == rxsum) {
185
               rcValue[0] = (ibus[3] << 8) + ibus[2];
rcValue[1] = (ibus[5] << 8) + ibus[4];
rcValue[2] = (ibus[7] << 8) + ibus[6];
187
188
               rcValue[3] = (ibus[9] << 8) + ibus[8];
               rcValue[4] = (ibus[11] << 8) + ibus[10];
rcValue[5] = (ibus[13] << 8) + ibus[12];
readingRx = false;
190
191
193
             }
        194
196
197
198
199
200
201
202
203
204
205
        3
206
207
      3
```

2/6

```
209
         readingRx = false:
         // if(micros()-loop_timer>700) PORTD ^= (1 << 7); while (micros() - loop_timer < 700);
210
                                                                                                                                      ////// End of Rx loop
       213
214
215
216
217
218
219
220
221
222
223
224
         loop_timer = micros();
                                                                                                                 // 1
        loop_timer = micros();
accelgyro.getMotion6(&ax, &ay, &az, &gx, &gx, &gz);
gyro_roll_meter = (gyro_roll_meter * 0.9) + (((gy - 28 ) / 10) * 0.1); //Gyro pid meter is deg/sec.
gyro_pitch_meter = (gyro_pitch_meter * 0.9) + (((gx + 343 ) / 10) * 0.1) * -1; //Gyro pid meter is deg/sec.
gyro_yaw_meter = (gyro_yaw_meter * 0.9) + (((gz + 234 ) / 10) * 0.1) * -1; //Gyro pid meter is deg/sec.
225
226
227
228
             Serial.print(gyro_roll_meter); Serial.print("\t");
229
230
231
              Serial.print(gyro_pitch_meter); Serial.print("\t");
         // Serial.println(gyro_yaw_meter);
232
233
        rc_command[0] = 0; // Roll
//We need a little dead band of 16us for better results.
234
235
         if (valueRoll > 1508 ){
236
        rc_command[0] = (valueRoll - 1508); // convert to degree per socond +500 //when roll to right +degree
} else if (valueRoll < 1492){
rc_command[0] = (valueRoll - 1492); // - 500</pre>
237
238
240
241
        rc_command[1] = 0; // Pitch
//We need a little dead band of 16us for better results. prevent RX noise
243
244
         if (valuePitch > 1508 ) {
        rc_command[1] = (valuePitch - 1508);
} else if (valuePitch < 1492) {</pre>
246
247
248
           rc_command[1] = (valuePitch - 1492);
249
250
251
         rc_yaw_command = 0; // Yaw
//We need a little dead band of 16us for better results.
if (valueThrottle > 1050) // only if the throttle is hight we can yaw!!!
252
253
         { //Do not yaw when turning off the motors.
if (valueYaw > 1508)
255
256
            rc_yaw_command = (valueYaw - 1508);
else if (valueYaw < 1492)
rc_yaw_command = (valueYaw - 1492);
258
259
260
        }
261
         valueThrottle = constrain(valueThrottle, 1010, 2000);
262
263
264
              Serial.print("a/g:\t");
              Serial.print(rc_command[0]); Serial.print("\t");
Serial.print(rc_command[1]); Serial.print("\t");
265
266
267
              Serial.println(rc_yaw_command);
268
269
         gyroInput[0] = gyro_roll_meter;
270
         gyroInput[1] = gyro_pitch_meter;
if (valueThrottle > 1000) {
271
272
273
            for (axis = \theta; axis < 2; axis++) {
274
                  PPPPPP
               pid_error_temp = rc_command[axis] * 4 - gyroInput[axis];
               Pterm = pid_error_temp * (float) pid_p_gain_roll / 150; // https://forum.arduino.cc/index.php?topic=28742.0
276
277
278
               // IIIIIII
279
              pid_i_error[axis] = constrain(pid_i_error[axis] + pid_error_temp, -16000, 16000);
if (abs(gyroInput[axis]) > 1500) pid_i_error[axis] = 0;
Iterm = pid_i_error[axis] * (float)pid_i_gain_roll / 3000 ; // pid_i_error[axis] * (pid_i_gain_roll / 100);
280
282
283
              // DDDDDD
285
286
                                     = gyroInput[axis] - lastgyro[axis];
              288
289
               delta2[axis]
                                     = delta1[axis];
291
              delta1[axis]
                                     = delta;
292
              Dterm = Dterm * (float)pid_d_gain_roll / 35;
294
              pidOutput[axis] = Pterm + Iterm - Dterm ;
295
296
297
            // Yaw pid calculation
298
           pid_error_temp = rc_yaw_command * 4 - gyro_yaw_meter;
pid_i_mem_yaw += pid_error_temp;
pid_i_mem_yaw = constrain(pid_i_mem_yaw, -16000, 1600
300
                                      constrain(pid i mem yaw, -16000, 16000);
301
           (abs(gyro_yaw_meter) > 650) && (pid_i_mem_yaw = 0);
// there is no d gain for yaw because no ocilation gravity
pid_output_yaw = (pid_error_temp * (float)pid_p_gain_yaw / 150) + (pid_i_mem_yaw * (float)pid_i_gain_yaw / 3000);
302
303
304
305
306
307
                    Serial.println( (pid_i_error[0]* pid_i_gain_roll) / 200)
         (micros() - loop_timer > 1500 && start == 1) && (PORTD ^= (1 << 7));
while (micros() - loop_timer < 1500) ;</pre>
309
                                                                                                                             ///////End of pid loop
310
       311
```

```
312
313
314
315
316
317
318
320
         // PORTD ^= (1<<7);
321
         PORTD |= B00001000;
PORTB |= B00001110;
322
                                       //Set digital outputs 3 high.
         PORTB |= B00001110; // set pin 9,10,11
loop_timer = micros(); // 1
323
324
326
           / Arming the quad
         if (valueThrottle < 1050 && valueAux1 > 1800 && start == 0) {
327
           PORTB |= B00100000;
328
329
           start = 1;
330
        }
332
         // un-arm the quad
         if (valueThrottle < 1050 && valueAux1 < 1200 && start >= 1 && valueAux2 < 1020) {
333
           PORTB &= B11011111;
           start = 0;
lastgyro[0] = 0;
335
336
            lastgyro[1] = 0;
           pid_i_error[0] = 0;
pid_i_error[1] = 0;
338
339
340
           pid_error_temp = 0;
341
           delta = 0:
342
           delta1[0] = 0;
delta1[1] = 0;
344
345
346
           delta2[0] = 0;
delta2[1] = 0;
347
348
350
351
         // set tune variable to pitch and roll
         /// if tuning is active set p i d key by button
// throught p i d key we can set pid gain base on that.
353
354
         if (valueThrottle < 1090 && valueAux2 > 1200 ) {
            if (valueAux2 < 1600 && (start == 0 || start == 4) ) {
356
              357
359
360
361
362
363
            if (valueAux2 > 1700 && start == 3) {
364
              readienus, 2 1700 & 31311 -- 3) {
start = 4;
p_light_k = EEPROM.read(6) * p_mul;
d_light_k = EEPROM.read(7) * d_mul;
i_light_k = EEPROM.read(8) * i_mul;
365
366
368
369
371
        }
372
                 Serial.print("\t");
        Serial.print(pid_p_gain_roll); Serial.print("\t");
Serial.print(pid_d_gain_roll); Serial.print("\t");
Serial.print(pid_i_gain_roll); Serial.print("\t");
374
375
376
                 Serial.print(pid_p_gain_yaw); Serial.print("\t");
Serial.print(pid_d_gain_yaw); Serial.print("\t");
377
378
                 Serial.println(pid_i_gain_yaw);
380
         if (valueThrottle < 1090 && valueAux2 < 1200 && start >= 3) {
381
383
           p_{light_k} = 0;
            d light k = 0:
384
            i_light_k = 0;
386
387
        }
389
         if (bitRead(PINC, 0) == 0 && button_pushed == 0)
390
         {
            // PORTD ^= (1 << 7);
(start == 3) && (pid_p_gain_roll += 1);
392
393
            (start == 4) && (pid_p_gain_yaw += 1);
394
395
            p light k += p \text{ mul}; // 1000/40 = 25
396
           p_light_k > Demol, // 1000/10 = 25
if (p_light_k > 1000) {
    p_light_k = 0;
    (start == 3) && (pid_p_gain_roll = 0);
398
399
              (start == 4) && (pid_p_gain_yaw = 0);
400
401
402
403
           button_pushed = 1;
        }
494
405
         if (bitRead(PINC, 2) == 0 && button_pushed == 0)
406
497
           // PORTD ^= (1 << 7);
(start == 3) && (pid_i_gain_roll += 1);
(start == 4) && (pid_i_gain_yaw += 1);
408
409
410
411
412
            i_light_k += i_mul; // 1000/40 = 25
           | 1_1ight_k = 1_midi, // 1000/40 = 25
| if (i_light_k = 0);
| (start == 3) && (pid_i_gain_roll = 0);
| (start == 4) && (pid_i_gain_yaw = 0);
413
414
```

```
417
418
         3
419
420
         button_pushed = 1;
421
       }
422
423
       if (bitRead(PINC, 1) == 0 && button_pushed == 0)
424
       ş
          // PORTD ^= (1 << 7);
(start == 3) && (pid_d_gain_roll += 1);
425
426
         (start == 4) && (pid_d_gain_yaw += 1);
427
428
429
          d light k += d \text{ mul}: // 1000/25 = 40
         if (d_light_k > 1000) {
430
           d_{light_k} = 0;
431
           (start == 3) && (pid_d_gain_roll = 0);
(start == 4) && (pid_d_gain_yaw = 0);
432
433
434
435
         }
436
437
         button_pushed = 1;
438
       }
439
440
441
       if (start >= 3 && valueYaw < 1090 && valuePitch < 1100 )
443
       {
         EEPROM.write(3, pid_p_gain_roll);
EEPROM.write(4, pid_d_gain_roll);
444
445
446
         EEPROM.write(5, pid_i_gain_roll);
447
448
          EEPROM.write(6, pid_p_gain_yaw);
449
         EEPROM.write(7, pid_d_gain_yaw)
         EEPROM.write(8, pid_i_gain_yaw);
450
451
          start = 2;
452
         PORTB |= B00100000;
       3
453
454
       455
456
       !!
!!
458
           Serial.print(EEPROM.read(4));
           Serial.print(EEPROM.read(5));
459
460
           Serial.print(EEPROM.read(6));
461
            Serial.print(EEPROM.read(7)
           Serial.println(EEPROM.read(8));
462
463
464
465
466
       //d5
/////// End of ESC command loop
467
468
469
       // cpp.sh/9nhqh // url used to learn the code
470
471
472
473
474
       if (start >= 3) {
    (p_light_k > 0) && (PORTD |= B10000000);
    (d_light_k > 0) && (PORTD |= B01000000);
475
476
477
478
          (i_light_k > 0) && (PORTD |= B00100000);
         PORTD &= B11110111:
                                                                                                        //Set digital output 10 to low if the time is expired.
479
          PORTB &= B11110001;
480
                                                                                                        //Set digital output 3 to low if the time is expired.
481
         loop_timer = micros();
         while (micros() - loop_timer < 1500 ) {
                                                                                                        //Stay in this loop until output all port are low number is
482
     bite
483
            esc_loop_timer = micros() - loop_timer;
            (esc_loop_timer >= p_light_k ) && (PORTD &= B01111111);
(esc_loop_timer >= d_light_k ) && (PORTD &= B10111111);
                                                                                                        // pin
484
                                                                                                        // pin
// pin
486
            (esc_loop_timer >= i_light_k ) && (PORTD &= B11011111);
487
         3
       }
488
489
490
497
493
494
       if (start <= 1 ) {
                                                                                                         // TODO: this will make esc beaping when tune pid
         loop_timer = micros();
while ((PIND & B00001000 || PINB & B00001110)) {
495
                                                                                                         //Stay in this loop until output all port are low number
496
497
           esc_loop_timer = micros();
           esc_loop_timer = micros();
if (esc_loop_timer - loop_timer >= esc_1 - 1000) PORTB &= B11111011;
if (esc_loop_timer - loop_timer >= esc_2 - 1000) PORTD &= B11110111;
if (esc_loop_timer - loop_timer >= esc_3 - 1000) PORTB &= B11110111;
if (esc_loop_timer - loop_timer >= esc_4 - 1000) PORTB &= B11111011;
                                                                                                         //Set digital output 10 to low if the time is expired.
498
                                                                                                         //Set digital output 3 to low if the time is expired.
499
                                                                                                         //Set digital output 11 to low if the time is expired.
500
                                                                                                         // pin 9
501
502
       }
503
504
     506
                                                                                                       //////End of motor loop
507
       while (micros() - loop_timer < 1600) ;
(button_pushed >= 1) && (button_pushed += 1);
508
509
       (button_pushed > 50) && (button_pushed = 0);
511
     // Total loop rx 700 + Gyro+pid 1500 + esc min up 1000+ esc to loow time 1600 = 4800
512
514
515
517
```