

1/6


```

209   readingRx = false;
210   // if(micros()-loop_timer>700) PORTD ^= (1 << 7);
211   while (micros() - loop_timer < 700); // End of Rx loop
212   ///////////////////////////////////
213
214
215
216
217
218
219
220
221
222
223
224   loop_timer = micros(); // 1
225   accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
226   gyro_roll_meter = (gyro_roll_meter * 0.9) + (((gy - 28) / 10) * 0.1); //Gyro pid meter is deg/sec.
227   gyro_pitch_meter = (gyro_pitch_meter * 0.9) + (((gx + 343) / 10) * 0.1) * -1; //Gyro pid meter is deg/sec.
228   gyro_yaw_meter = (gyro_yaw_meter * 0.9) + (((gz + 234) / 10) * 0.1) * -1; //Gyro pid meter is deg/sec.
229   // Serial.print("a/g:\t");
230   // Serial.print(gyro_roll_meter); Serial.print("\t");
231   // Serial.print(gyro_pitch_meter); Serial.print("\t");
232   // Serial.println(gyro_yaw_meter);
233
234   rc_command[0] = 0; // Roll
235   //We need a little dead band of 16us for better results.
236   if (valueRoll > 1508) {
237     rc_command[0] = (valueRoll - 1508); // convert to degree per socond +500 //when roll to right +degree
238   } else if (valueRoll < 1492) {
239     rc_command[0] = (valueRoll - 1492); // - 500
240   }
241
242
243   rc_command[1] = 0; // Pitch
244   //We need a little dead band of 16us for better results. prevent RX noise
245   if (valuePitch > 1508) {
246     rc_command[1] = (valuePitch - 1508);
247   } else if (valuePitch < 1492) {
248     rc_command[1] = (valuePitch - 1492);
249   }
250
251
252   rc_yaw_command = 0; // Yaw
253   //We need a little dead band of 16us for better results.
254   if (valueThrottle > 1050) // only if the throttle is hight we can yaw!!!
255   { //Do not yaw when turning off the motors.
256     if (valueYaw > 1508)
257       rc_yaw_command = (valueYaw - 1508);
258     else if (valueYaw < 1492)
259       rc_yaw_command = (valueYaw - 1492);
260   }
261
262   valueThrottle = constrain(valueThrottle, 1010, 2000);
263
264   // Serial.print("a/g:\t");
265   // Serial.print(rc_command[0]); Serial.print("\t");
266   // Serial.print(rc_command[1]); Serial.print("\t");
267   // Serial.println(rc_yaw_command);
268
269   gyroInput[0] = gyro_roll_meter;
270   gyroInput[1] = gyro_pitch_meter;
271   if (valueThrottle > 1000) {
272
273     for (axis = 0; axis < 2; axis++) {
274       // P P P P P
275       pid_error_temp = rc_command[axis] * 4 - gyroInput[axis];
276       Pterm = pid_error_temp * (float) pid_p_gain_roll / 150; // https://forum.arduino.cc/index.php?topic=28742.0
277
278
279       // I I I I I I I
280       pid_i_error[axis] = constrain(pid_i_error[axis] + pid_error_temp, -16000, 16000);
281       if (abs(gyroInput[axis]) > 1500) pid_i_error[axis] = 0;
282       Iterm = pid_i_error[axis] * (float) pid_i_gain_roll / 3000; // pid_i_error[axis] * (pid_i_gain_roll / 100);
283
284
285       // D D D D D
286
287       delta = gyroInput[axis] - lastgyro[axis];
288       lastgyro[axis] = gyroInput[axis];
289       Dterm = delta + delta1[axis] + delta2[axis];
290       delta2[axis] = delta1[axis];
291       delta1[axis] = delta;
292
293       Dterm = Dterm * (float) pid_d_gain_roll / 35;
294
295       pidOutput[axis] = Pterm + Iterm - Dterm;
296     }
297
298     // Yaw pid calculation
299     pid_error_temp = rc_yaw_command * 4 - gyro_yaw_meter;
300     pid_i_mem_yaw += pid_error_temp;
301     pid_i_mem_yaw = constrain(pid_i_mem_yaw, -16000, 16000);
302     (abs(gyro_yaw_meter) > 650) && (pid_i_mem_yaw = 0);
303     // there is no d gain for yaw because no ocilation gravity
304     pid_output_yaw = (pid_error_temp * (float) pid_p_gain_yaw / 150) + (pid_i_mem_yaw * (float) pid_i_gain_yaw / 3000);
305   }
306
307
308   // Serial.println( (pid_i_error[0]* pid_i_gain_roll) / 200);
309   (micros() - loop_timer > 1500 && start == 1) && (PORTD ^= (1 << 7));
310   while (micros() - loop_timer < 1500); // End of pid loop
311   ///////////////////////////////////

```

```

312
313
314
315
316
317
318
319
320
321 // PORTD ^= (1<<7);
322 PORTD |= B00001000; //Set digital outputs 3 high.
323 PORTB |= B00001110; // set pin 9,10,11
324 loop_timer = micros(); // 1
325
326 // Arming the quad
327 if (valueThrottle < 1050 && valueAux1 > 1800 && start == 0) {
328     PORTB |= B00100000;
329     start = 1;
330 }
331
332 // un-arm the quad
333 if (valueThrottle < 1050 && valueAux1 < 1200 && start >= 1 && valueAux2 < 1020) {
334     PORTB &= B11011111;
335     start = 0;
336     lastgyro[0] = 0;
337     lastgyro[1] = 0;
338     pid_i_error[0] = 0;
339     pid_i_error[1] = 0;
340     pid_error_temp = 0;
341
342     delta = 0;
343
344     delta1[0] = 0;
345     delta1[1] = 0;
346
347     delta2[0] = 0;
348     delta2[1] = 0;
349 }
350
351
352 // set tune variable to pitch and roll
353 // if tuning is active set p i d key by button
354 // throught p i d key we can set pid gain base on that.
355 if (valueThrottle < 1090 && valueAux2 > 1200 ) {
356     if (valueAux2 < 1600 && (start == 0 || start == 4) ) {
357         start = 3;
358         p_light_k = EEPROM.read(3) * p_mul; // 1*50 = 50, 2*50 = 100 ; P : 1/10=0.1 ; 2/10=0.2;
359         d_light_k = EEPROM.read(4) * d_mul; // 2*25 = 50, 4*25 = 100 ; D : 1/10=0.1 ; 2/10=0.2;
360         i_light_k = EEPROM.read(5) * i_mul; // 1*50 = 50, 2*50 = 100 I : 1/100 = 0.01
361
362     }
363
364     if (valueAux2 > 1700 && start == 3) {
365         start = 4;
366         p_light_k = EEPROM.read(6) * p_mul;
367         d_light_k = EEPROM.read(7) * d_mul;
368         i_light_k = EEPROM.read(8) * i_mul;
369     }
370 }
371
372
373 // Serial.print("\t");
374 // Serial.print(pid_p_gain_roll); Serial.print("\t");
375 // Serial.print(pid_d_gain_roll); Serial.print("\t");
376 // Serial.print(pid_i_gain_roll); Serial.print("\t");
377 // Serial.print(pid_p_gain_yaw); Serial.print("\t");
378 // Serial.print(pid_d_gain_yaw); Serial.print("\t");
379 // Serial.println(pid_i_gain_yaw);
380
381 if (valueThrottle < 1090 && valueAux2 < 1200 && start >= 3) {
382     start = 0;
383     p_light_k = 0;
384     d_light_k = 0;
385     i_light_k = 0;
386
387 }
388
389
390 if (bitRead(PINC, 0) == 0 && button_pushed == 0)
391 {
392     // PORTD ^= (1 << 7);
393     (start == 3) && (pid_p_gain_roll += 1);
394     (start == 4) && (pid_p_gain_yaw += 1);
395
396     p_light_k += p_mul; // 1000/40 = 25
397     if (p_light_k > 1000) {
398         p_light_k = 0;
399         (start == 3) && (pid_p_gain_roll = 0);
400         (start == 4) && (pid_p_gain_yaw = 0);
401     }
402
403     button_pushed = 1;
404 }
405
406 if (bitRead(PINC, 2) == 0 && button_pushed == 0)
407 {
408     // PORTD ^= (1 << 7);
409     (start == 3) && (pid_i_gain_roll += 1);
410     (start == 4) && (pid_i_gain_yaw += 1);
411
412     i_light_k += i_mul; // 1000/40 = 25
413     if (i_light_k > 1000) {
414         i_light_k = 0;
415         (start == 3) && (pid_i_gain_roll = 0);
416         (start == 4) && (pid_i_gain_yaw = 0);

```

```

417     }
418
419     button_pushed = 1;
420 }
421
422 if (bitRead(PINC, 1) == 0 && button_pushed == 0)
423 {
424     // PORTD ^= (1 << 7);
425     (start == 3) && (pid_d_gain_roll += 1);
426     (start == 4) && (pid_d_gain_yaw += 1);
427
428     d_light_k += d_mul; // 1000/25 = 40
429     if (d_light_k > 1000) {
430         d_light_k = 0;
431         (start == 3) && (pid_d_gain_roll = 0);
432         (start == 4) && (pid_d_gain_yaw = 0);
433     }
434
435     button_pushed = 1;
436 }
437
438 if (start >= 3 && valueYaw < 1090 && valuePitch < 1100 )
439 {
440     EEPROM.write(3, pid_p_gain_roll);
441     EEPROM.write(4, pid_d_gain_roll);
442     EEPROM.write(5, pid_i_gain_roll);
443
444     EEPROM.write(6, pid_p_gain_yaw);
445     EEPROM.write(7, pid_d_gain_yaw);
446     EEPROM.write(8, pid_i_gain_yaw);
447     start = 2;
448     PORTB |= B00100000;
449 }
450
451 //////////////////////////////////////////////////End of pid command////////////////////////////////////
452 // Serial.print(start);
453 // Serial.print(EEPROM.read(3));
454 // Serial.print(EEPROM.read(4));
455 // Serial.print(EEPROM.read(5));
456 // Serial.print(EEPROM.read(6));
457 // Serial.print(EEPROM.read(7));
458 // Serial.println(EEPROM.read(8));
459
460 (micros() - loop_timer > 950 && start == 1) && (PORTD ^= (1 << 5)); //d5
461 while (micros() - loop_timer <= 1000); //End of ESC command loop
462 //////////////////////////////////////////////////
463 // cpp.sh/9nhqh // url used to learn the code
464
465 if (start >= 3) {
466     (p_light_k > 0) && (PORTD |= B10000000);
467     (d_light_k > 0) && (PORTD |= B01000000);
468     (i_light_k > 0) && (PORTD |= B00100000);
469     PORTD &= B11110111; //Set digital output 10 to low if the time is expired.
470     PORTB &= B11110001; //Set digital output 3 to low if the time is expired.
471     loop_timer = micros();
472     while (micros() - loop_timer < 1500) { //Stay in this loop until output all port are low number is
473         bite
474             esc_loop_timer = micros() - loop_timer;
475             (esc_loop_timer >= p_light_k ) && (PORTD &= B01111111); // pin
476             (esc_loop_timer >= d_light_k ) && (PORTD &= B10111111); // pin
477             (esc_loop_timer >= i_light_k ) && (PORTD &= B11011111); // pin
478         }
479     }
480
481 if (start <= 1 ) { // TODO: this will make esc beeping when tune pid
482     loop_timer = micros(); // 2
483     while ((PIND & B00001000 || PINB & B00001110)) { //Stay in this loop until output all port are low number
484         is bite
485             esc_loop_timer = micros();
486             if (esc_loop_timer - loop_timer >= esc_1 - 1000) PORTB &= B11110111; //Set digital output 10 to low if the time is expired.
487             if (esc_loop_timer - loop_timer >= esc_2 - 1000) PORTD &= B11110111; //Set digital output 3 to low if the time is expired.
488             if (esc_loop_timer - loop_timer >= esc_3 - 1000) PORTB &= B11110111; //Set digital output 11 to low if the time is expired.
489             if (esc_loop_timer - loop_timer >= esc_4 - 1000) PORTB &= B11111011; // pin 9
490         }
491     }
492
493 (micros() - loop_timer > 1600 && start == 1) && (PORTD ^= (1 << 6)); // D6
494 //////////////////////////////////////////////////End of motor loop
495 //Total loop rx 700 + Gyro+pid 1500 + esc min up 1000+ esc to loow time 1600 = 4800
496 while (micros() - loop_timer < 1600 );
497 (button_pushed >= 1) && (button_pushed += 1);
498 (button_pushed > 50) && (button_pushed = 0);
499
500 // Total loop rx 700 + Gyro+pid 1500 + esc min up 1000+ esc to loow time 1600 = 4800
501
502 }
503
504 }
505
506 }
507
508 }
509
510 }
511
512 }
513
514 }
515
516 }
517

```

