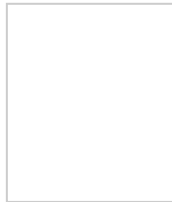




(<http://radioparty.ru>).



[Главная \(/\)](#) / [Программирование](#) / [AVR](#)

/ [Программирование на СИ \(/programming/avr/c?filter\\_tag\[0\]=\)](#)

/ [Управление бесколлекторным двигателем постоянного тока\(IR2101\)](#)

# Управление бесколлекторным двигателем постоянного тока(IR2101)

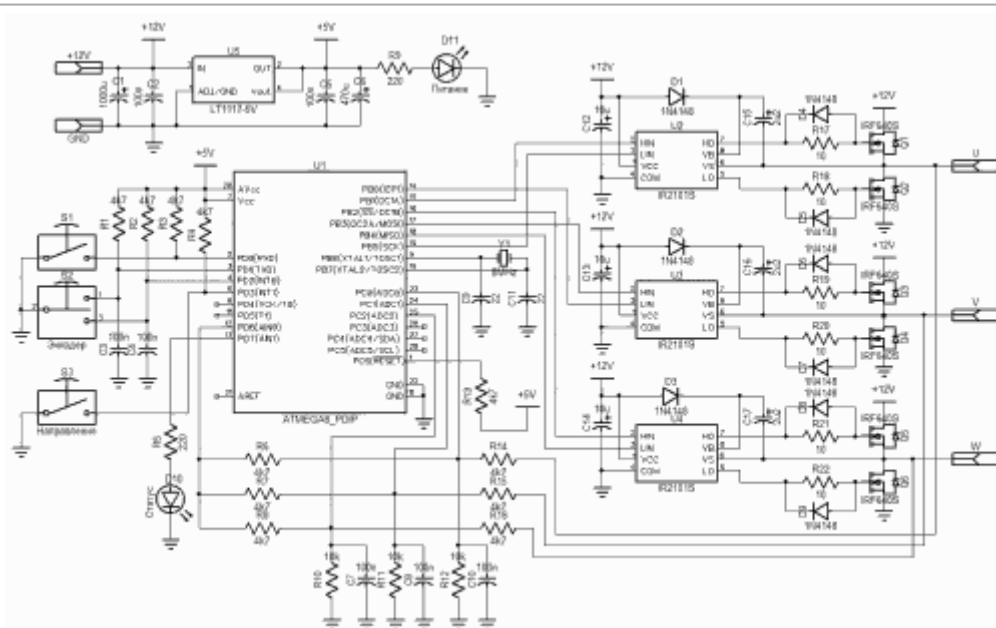
Автор: [AntonChip \(/contact\)](#). Дата публикации: 15 февраля 2017.



Оценка 5

Оценить

В этом варианте контроллера бесколлекторного двигателя для управления силовыми ключами используется специальный драйвер IR2101, который позволяет избавиться от дорогих и дефицитных Р-канальных полевых транзисторов. В исходный код внес некоторые изменения, в частности плавный пуск и изменение вращения вала двигателя. При первом запуске двигатель плавно разгоняется, от уровня ШИМ START\_PWM, до уровня WORK\_PWM, задержку можно поменять. При повторном запуске двигатель разгонится уже до уровня, который будет в переменной motor\_pwm. Эта переменная меняется при помощи ручки энкодера. Переключатель направления вращения работает только когда двигатель находится в выключенном состоянии.



## Исходный код программы:

```
1 // Подключение бесколлекторного двигателя к AVR(без датчиков)
2 #include <avr/interrupt.h>
3 #include <avr/io.h>
4 #include <util/delay.h>
5
6 // Фаза U(Верхнее плечо)
7 #define UH_ON TCCR1A |= (1 << COM1A1);
8 #define UH_OFF TCCR1A &= ~(1 << COM1A1);
9
10 // Фаза U(Нижнее плечо)
11 #define UL_ON PORTB |= (1 << PB5);
12 #define UL_OFF PORTB &= ~(1 << PB5);
13
14 // Фаза V(Верхнее плечо)
15 #define VH_ON TCCR2 |= (1 << COM21);
16 #define VH_OFF TCCR2 &= ~(1 << COM21);
17
18 // Фаза V(Нижнее плечо)
19 #define VL_ON PORTB |= (1 << PB0);
20 #define VL_OFF PORTB &= ~(1 << PB0);
21
22 // Фаза W(Верхнее плечо)
23 #define WH_ON TCCR1A |= (1 << COM1B1);
24 #define WH_OFF TCCR1A &= ~(1 << COM1B1);
25
26 // Фаза W(Нижнее плечо)
27 #define WL_ON PORTB |= (1 << PB4);
28 #define WL_OFF PORTB &= ~(1 << PB4);
29
30 #define PHASE_ALL_OFF UH_OFF;UL_OFF;VH_OFF;VL_OFF;WH_OFF;WL_OFF;
31
32 #define SENSE_U ADMUX = 0; // Вход обратной ЭДС фазы U
33 #define SENSE_V ADMUX = 1; // Вход обратной ЭДС фазы V
34 #define SENSE_W ADMUX = 2; // Вход обратной ЭДС фазы W
35
36 #define SENSE_UVW (ACSR&(1 << ACO)) // Выход компаратора
37
38 #define START_PWM 10 // Минимальный ШИМ при запуске
39 #define WORK_PWM 100 // Максимальный уровень ШИМ при запуске
40
41 unsigned char start = 0, start_stop = 0, start_pwm;
42 unsigned char direction = 1; // 0 - против часовой, 1 - по часовой
43 volatile unsigned char motor_pwm = WORK_PWM;
44 volatile unsigned char commutation_step = 0;
45 volatile unsigned char rotor_run = 0; // Счетчик импульсов обратной ЭДС
46
47 // Функция переключения обмоток двигателя
48 void commutation(unsigned char startup)
49 {
50     switch (commutation_step)
51     {
52     case (0):
53         if(!SENSE_UVW || startup)
54         {
55             if(direction)
56             {
57                 UH_ON; WH_OFF; SENSE_W;
58             }
59             else
60             {
61                 UH_OFF; WH_ON; SENSE_U;
62             }
63             commutation_step = 1; // Следующий шаг
64             TCNT0 = 0; // Обнуляем счетчик T0
65         }
66         break;
67     }
```

```

68     case (1):
69         if(SENSE_UVW || startup)
70         {
71             if(direction)
72             {
73                 VL_OFF; WL_ON; SENSE_V;
74             }
75             else
76             {
77                 VL_OFF; UL_ON; SENSE_V;
78             }
79             commutation_step = 2;
80             TCNT0 = 0; // Обнуляем счетчик T0
81         }
82         break;
83
84     case (2):
85         if(!SENSE_UVW || startup)
86         {
87             if(direction)
88             {
89                 UH_OFF; VH_ON; SENSE_U;
90             }
91             else
92             {
93                 VH_ON; WH_OFF; SENSE_W;
94             }
95             commutation_step = 3;
96             TCNT0 = 0; // Обнуляем счетчик T0
97         }
98         break;
99
100    case (3):
101        if(SENSE_UVW || startup)
102        {
103            if(direction)
104            {
105                UL_ON; WL_OFF; SENSE_W;
106            }
107            else
108            {
109                UL_OFF; WL_ON; SENSE_U;
110            }
111            commutation_step = 4;
112            TCNT0 = 0; // Обнуляем счетчик T0
113        }
114        break;
115
116    case (4):
117        if(!SENSE_UVW || startup)
118        {
119            if(direction)
120            {
121                VH_OFF; WH_ON; SENSE_V;
122            }
123            else
124            {
125                VH_OFF; UH_ON; SENSE_V;
126            }
127            commutation_step = 5;
128            TCNT0 = 0; // Обнуляем счетчик T0
129        }
130        break;
131
132    case (5):
133        if(SENSE_UVW || startup)
134        {
135            if(direction)
136            {
137                UL_OFF; VL_ON; SENSE_U;
138            }
139            else

```

```

140         {
141             VL_ON; WL_OFF; SENSE_W;
142         }
143         commutation_step = 0;
144         TCNT0 = 0; // Обнуляем счетчик T0
145     }
146     break;
147 }
148 }
149 // Обработчик прерывания по компаратору. Детектор обратной ЭДС
150 ISR(ANA_COMP_vect)
151 {
152     rotor_run++; // инкрементируем импульсы
153     if(rotor_run > 200) rotor_run = 200;
154     if(rotor_run == 200) // Если импульсы обратной ЭДС присутствуют, крутим на полную
155     commutation(0); // Переключаем обмотки по сигналу компаратора
156 }
157 // Обработчик прерывания по переполнению T0. Работа двигателя без сигналов обратной ЭДС
158 // Если сработало прерывание, есть пропуски импульсов обратной ЭДС
159 ISR(TIMER0_OVF_vect)
160 {
161     rotor_run = 0; // Сбрасываем счетчик импульсов
162     OCR1A = START_PWM; // ШИМ минимум
163     OCR1B = START_PWM;
164     OCR2 = START_PWM;
165     commutation(1); // Переключаем обмотки безусловно
166 }
167 // Обработчик внешнего прерывания INT0. Энкодер
168 ISR(INT0_vect){
169     _delay_us(100);
170     if ((PIND & ( 1 << PD2)) == 0){
171         _delay_us(100);
172         // Крутим против часовой стрелки
173         if ((PIND & ( 1 << PD1)) == 0)
174         {
175             if(motor_pwm != START_PWM) motor_pwm -= 5; // Уменьшаем ШИМ
176         }
177         // Крутим по часовой стрелке
178         else
179         {
180             if(motor_pwm != 255) motor_pwm += 5; // Увеличиваем ШИМ
181         }
182     }
183     GIFR = (1 << INTF0); // Сбрасываем флаг внешнего прерывания
184     return;
185 }
186
187 int main (void)
188 {
189     // Порты ввода/вывода
190     DDRB = 0xFF;
191     PORTB = 0x00;
192     DDRD |= (1 << PD7);
193     DDRD &= ~(1 << PD6)|(1 << PD3)|(1 << PD2)|(1 << PD1)|(1 << PD0);
194     PORTD |= (1 << PD3)|(1 << PD2)|(1 << PD1)|(1 << PD0);
195     PORTD &= ~(1 << PD7)|(1 << PD6);
196
197     // T0 - для старта и работы двигателя без сигналов обратной ЭДС
198     TCCR0 |= (1 << CS02)|(1 << CS00); // Предделитель на 1024
199     TIMSK |= (1 << TOIE0); // Разрешаем прерывание по переполнению T0
200     // T1 и T2 ШИМ
201     TCCR1A |= (1 << COM1A1)|(1 << COM1B1) // Clear OC1A/OC1B, set OC1A/OC1B at BOTTOM
202             (1 << WGM10); // Режим Fast PWM, 8-bit
203     TCCR1B |= (1 << CS10)|(1 << WGM12); // Без предделителя
204     TCCR2 |= (1 << COM21) // Clear OC2, set OC2 at BOTTOM
205             (1 << WGM21)|(1 << WGM20) // Режим Fast PWM
206             (1 << CS20); // Без предделителя
207
208     PHASE_ALL_OFF; // Выключаем все фазы
209
210     // Аналоговый компаратор
211     ADCSRA &= ~(1 << ADEN); // Выключаем АЦП

```

```

212 SFIOR |= (1 << ACME); // Отрицательный вход компаратора подключаем к выходу мульт
213 ACSR |= (1 << ACIE); // Разрешаем прерывания от компаратора
214
215 // Внешнее прерывание(Энкодер)
216 MCUCR |= (1 << ISC01); // Прерывание по заднему фронту INT0(по спаду импульса)
217 GIFR |= (1 << INTF0); // Очищаем флаг внешнего прерывания
218 GICR |= (1 << INT0); // Разрешаем внешние прерывания INT0
219
220 sei(); // Глобально разрешаем прерывания
221
222 while(1)
223 {
224     if((PIND&(1 << PD0)) == 0) // Старт/Стоп
225     {
226         _delay_ms(20);
227         start_stop ^= 1; // Переключаем состояние
228         while((PIND&(1 << PD0)) == 0){} // Ждем отпускания кнопки
229     }
230
231     if(start_stop)
232     {
233         ACSR |= (1 << ACIE); // Разрешаем прерывание от компаратора
234         TIMSK |= (1 << TOIE0); // Разрешаем прерывание по переполнению T0
235         GICR |= (1 << INT0); // Разрешаем внешние прерывания INT0
236         // Плавный старт
237         if(rotor_run == 200 && start == 0) // Если импульсы обратной ЭДС присутствуют
238         {
239             for(start_pwm = START_PWM; start_pwm < motor_pwm; start_pwm++)
240             {
241                 _delay_ms(10); // Задержка
242                 OCR1A = start_pwm;
243                 OCR1B = start_pwm;
244                 OCR2 = start_pwm;
245             }
246             start = 1; // Запуск произошел
247             PORTD |= (1 << PD7); // Включаем светодиод
248         }
249
250         if(rotor_run == 200) // Если импульсы обратной ЭДС присутствуют, можем менять
251         {
252             OCR1A = motor_pwm;
253             OCR1B = motor_pwm;
254             OCR2 = motor_pwm;
255         }
256     }
257     else
258     {
259
260         if(PIND&(1 << PD3)) direction = 1; // Выбор направления вращения вала
261         else direction = 0;
262
263         start = 0; // Двигатель остановлен
264         PORTD &= ~(1 << PD7); // Выключаем светодиод
265         PHASE_ALL_OFF; // Все фазы выключены
266         ACSR &= ~(1 << ACIE); // Запрещаем прерывание от компаратора
267         TIMSK &= ~(1 << TOIE0); // Запрещаем прерывание по переполнению T0
268         GICR &= ~(1 << INT0); // Запрещаем внешние прерывания INT0
269     }
270
271 }
272 }

```



(</images/jdownloads/fileimages/rar.png>). **Архив для статьи**

## "Управление бесколлекторным двигателем постоянного тока(IR2101)" HOT

Проект AVRStudio4, схема, макет печатной платы DipTrace



(</images/jdownloads/miniimages/stuff.png>).

38.76 KB

Download  
(</component/jdownloads/send/2-downloads-avr/613-632>)



(</images/jdownloads/miniimages/download.png>).

524

2

Метки: [ATmega8 \(/device/tags/atmega8\)](/device/tags/atmega8), [Бесколлекторный двигатель \(/device/tags/beskollektornyj-dvigatel\)](/device/tags/beskollektornyj-dvigatel), [IR2101 \(/device/tags/ir2101\)](/device/tags/ir2101), [Датчик Холла \(/device/tags/datchik-kholla\)](/device/tags/datchik-kholla).

[Печать \(/programming/avr/c/632-lesson-blcd-sensorless-ir2101?tmpl=component&print=1&layout=default\)](/programming/avr/c/632-lesson-blcd-sensorless-ir2101?tmpl=component&print=1&layout=default). [E-mail \(/component/mailto/?tmpl=component&template=jp-b2b&link=9fe2cf8ae4a70d9131065ddb5674a3af019fb6b7\)](/component/mailto/?tmpl=component&template=jp-b2b&link=9fe2cf8ae4a70d9131065ddb5674a3af019fb6b7).

« Назад (</programming/avr/c/611-lesson-blcd-sensorless>)

Вперёд » (</programming/avr/c/636-lesson-blcd-sensorless-ir2101-hall-sensor>)

## Комментарии

1

2

#21 (</programming/avr/c/632-lesson-blcd-sensorless-ir2101#comment-3792>)

0

**Сергей V** 16.12.2017 18:38

Здравствуйте , а можете сказать до каких максимальных оборотов работает двигатель в вашей схеме. Давно искал такое решение для управления движком.

[Сообщить модератору](#)

#22 (</programming/avr/c/632-lesson-blcd-sensorless-ir2101#comment-3797>)

0

**AntonChip** 19.12.2017 08:16

Цитирую Сергей V:

*Здравствуйте , а можете сказать до каких максимальных оборотов работает двигатель в вашей схеме. Давно искал такое решение для управления движком.*

Все зависит от характеристик двигателя, подаваемого напряжения, мой двигатель крутит на максимальных 2500 об/мин, а например двигатель от HDD будет тоже вращаться на макс оборотах

[Сообщить модератору](#)

#23 (</programming/avr/c/632-lesson-blcd-sensorless-ir2101#comment-3801>)

0

**Сергей V** 21.12.2017 16:36

А у Вас случайно не найдется прошивка в формате hex ?

[Сообщить модератору](#)

1

2

[Обновить список комментариев](#)



















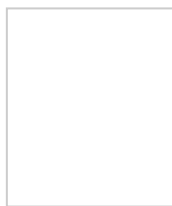












© 2010-2018 Микроконтроллеры и Технологии. Все права защищены.  
Материалы представленные на сайте исключительно для ознакомления.



([http://top.mail.ru/jump?  
from=1775061](http://top.mail.ru/jump?from=1775061))



([http://top100.rambler.ru/home?  
id=1979438](http://top100.rambler.ru/home?id=1979438))



([http://radionet.com.ru/top/c.php?  
id=7158](http://radionet.com.ru/top/c.php?id=7158))