

г. Караганда, ул. Алиханова 37, офис 108
г. Алматы, ул. Байтурсынова 85, блок Г,
офис 11
г. Астана, проспект Абая, 24/1, офис 47

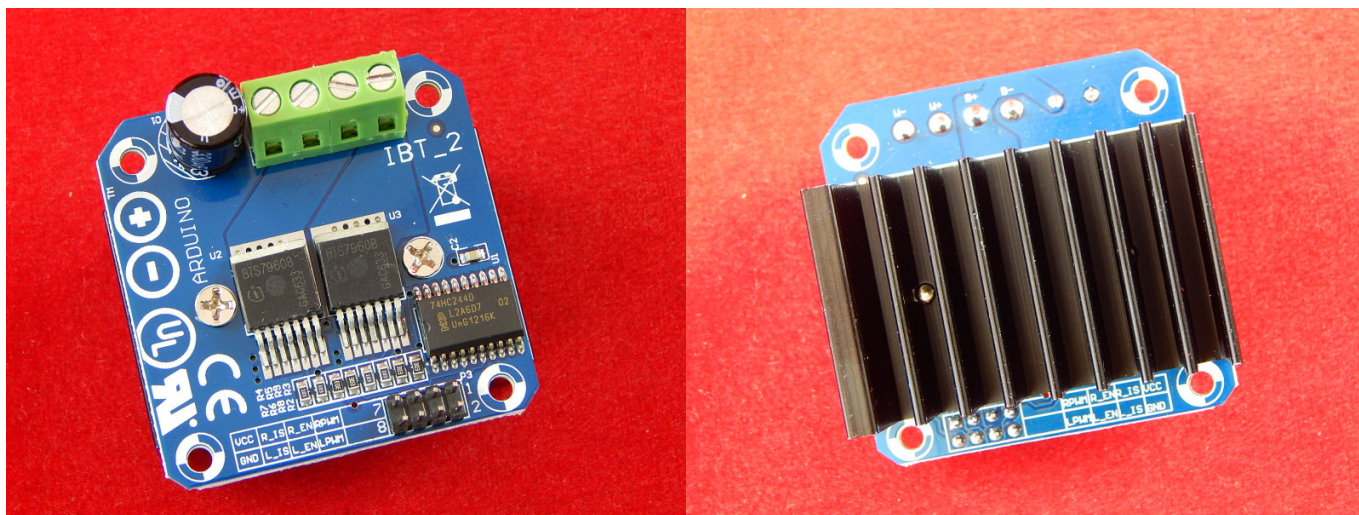
E-Mail: support@radiomart.org



Артикул: 14542

Цена в прайсе: 2861 тг.

Драйвер мотора на 43А BTS7960 для Arduino



С помощью данного драйвера можно управлять скоростью двигателя, направлением его движения, выполнять торможение и контролировать нагрузку которую испытывает двигатель.

Драйвер мотора BTS7960 — позволяет управлять одним коллекторным двигателем рассчитанным на напряжение от 5,5 до 27,5 В постоянного тока до 43 А. Но так как клеммник установленный на плате не рассчитан на такие токи, то для долговременной работы рекомендуется использовать двигатели с током потребления до 10 А.

Характеристики:

- Напряжение питания двигателя: от 5,5 до 27,5 В постоянного тока (вне указанного диапазона драйвер перейдет в режим защиты).
- Напряжение питания логики: 5 В постоянного тока.
- Максимально допустимы ток мотора: 43 А (кратковременно).
- Максимально допустимы ток мотора: 10 А (долговременно).
- Максимальная частота ШИМ на управляющих выводах: 25 кГц
- Напряжение логических уровней на управляющих выводах: 3,3 или 5 В

Подключение

Назначение выводов

- «M+» и «M-» - Выходы для подключения мотора.
- «M-» Выход левого плеча H-моста (чип U3).
- «M+» Выход правого плеча H-моста (чип U2).
- «S+» и «S-» - Питание мотора.
- «Vcc» и «GND» - Питание логики.

- «L_IS» - Выход состояния левого плеча Н-моста (чип U3).
Выход является флагом ошибки (если она есть), иначе с этого выхода можно снимать уровень напряжения пропорциональный току протекающему через мотор, а значит контролировать силу нагрузки приложенной к мотору.
- «R_IS» - Выход состояния правого плеча Н-моста (чип U2).
Выход является флагом ошибки (если она есть), иначе с этого выхода можно снимать уровень напряжения пропорциональный току протекающему через мотор, а значит контролировать силу нагрузки приложенной к мотору.
- «L_EN» - Вход разрешения работы левого плеча Н-моста (чип U3).
Сброс в 0 - отключает выход мотора «М-» (переводит его в состояние высокого импеданса).
Установка в 1 - разрешает работу выхода мотора «М-».
- «R_EN» - Вход разрешения работы правого плеча Н-моста (чип U2).
Сброс в 0 - отключает выход мотора «М+» (переводит его в состояние высокого импеданса).
Установка в 1 - разрешает работу выхода мотора «М+».
- «L_PWM» - Вход установки потенциала на выходе левого плеча Н-моста (чип U3).
Сброс в 0 - устанавливает на выходе «М-» потенциал вывода «S-».
Установка в 1 - устанавливает на выходе «М-» потенциал вывода «S+».
Установка потенциалов «S+» или «S-» на выходе «М-» возможна только если установлена 1 на входе «L_EN».
- «R_PWM» - Вход установки потенциала на выходе правого плеча Н-моста (чип U2).
Сброс в 0 - устанавливает на выходе «М+» потенциал вывода «S-».
Установка в 1 - устанавливает на выходе «М+» потенциал вывода «S+».
Установка потенциалов «S+» или «S-» на выходе «М+» возможна только если установлена 1 на входе «R_EN»

Подключение драйвера

- Двигатель подключается к выводам «М+» и «М-» клеммника.
- Напряжение питания двигателя (5,5 - 27,5 В постоянного тока) подаётся на выводы «S+» и «S-» клеммника.
- Напряжение питания логической части (5 В постоянного тока) подаётся на выводы «Vcc» и «GND».
- Драйвером можно управлять по 2, 3, или 4 проводам:

Подключение драйвера мотора к Arduino по 2 проводам:

Выходы «L_EN» и «R_EN» драйвера соединяются друг с другом и подключаются к «Vcc» (не участвуют в управлении).

Выход «L_PWM» подключается к любому выводу Arduino поддерживающему ШИМ.

Выход «R_PWM» подключается к любому выводу Arduino поддерживающему ШИМ.

Управление драйвером мотора по 2 проводам:

Движение вперёд с регулировкой скорости: «L_PWM» = 0, «R_PWM» = ШИМ (чем выше ШИМ тем выше скорость).

Движение вперёд с регулировкой скорости: «L_PWM» = ШИМ, «R_PWM» = 1 (чем выше ШИМ тем ниже скорость).

Движение назад с регулировкой скорости: «L_PWM» = ШИМ, «R_PWM» = 0 (чем выше ШИМ тем выше скорость).

Движение назад с регулировкой скорости: «L_PWM» = 1, «R_PWM» = ШИМ (чем выше ШИМ тем ниже скорость).

Торможение: «L_PWM» = «R_PWM» = 0 или 1 (максимальное торможение).

Преимущества и недостатки схемы управления по 2 проводам:

Явное преимущество схемы - минимальное количество задействованных выводов Arduino.

Так как на входах «L_EN» и «R_EN» постоянно установлена 1 (они подключены к «Vcc»), значит выходы мотора «M+» и «M-» не переходят в состояние высокого импеданса (не отключаются), следовательно, можно осуществлять торможение скоростью (уменьшение скорости приводит к торможению). Это-же факт можно отнести и к недостаткам схемы, так как в ней нельзя освободить мотор, на его выводах «M+» и «M-» всегда установлены потенциалы «S+» и/или «S-».

Подключение драйвера мотора к Arduino по 3 проводам:

Выводы «L_EN» и «R_EN» драйвера соединяются друг с другом и подключаются к любому выводу Arduino поддерживающему ШИМ.

Вывод «L_PWM» подключается к любому выводу Arduino.

Вывод «R_PWM» подключается к любому выводу Arduino.

Управление драйвером мотора по 3 проводам:

Движение вперед с регулировкой скорости: «L_PWM» = 0, «R_PWM» = 1, «EN» = ШИМ (чем выше ШИМ тем выше скорость).

Движение назад с регулировкой скорости: «L_PWM» = 1, «R_PWM» = 0, «EN» = ШИМ (чем выше ШИМ тем выше скорость).

Свободное вращение: «L_PWM» и «R_PWM» не имеют значения, «EN» = 0 (мотор электрически отключён).

Торможение: «L_PWM» = «R_PWM» = 0 или 1, «EN» = ШИМ (чем выше ШИМ тем сильнее торможение).

Преимущества и недостатки схемы управления по 3 проводам:

Не смотря на большее количество проводов, схема управления выглядит проще: «L_PWM» и «R_PWM» управляют направлением, а «EN» скоростью. Если на входах «L_PWM» и «R_PWM» одинаковый логический уровень, то «EN» управляет торможением.

Есть возможность регулировать уровень торможения при помощи ШИМ без подачи напряжения (разности потенциалов) на двигатель.

При подаче логического 0 на вход «EN», мотор электрически отключается от схемы.

Например, если управляемое мотором устройство стоит на горке и на всех входах «L_PWM», «R_PWM» и «EN» установлена 1, то оно никуда не поедет, но стоит сбросить в 0 уровень на входе «EN», как мотор освободится и устройство скатится с горки. Еще одним примером может быть экономия электроэнергии: после достижения требуемой скорости, сбрасываем уровень на входе «EN» в состояние логического 0 и устройство будет продолжать движение по инерции, далее устанавливаем на входе «EN» логическую 1, набираем скорость и снова сбрасываем в 0. К недостаткам схемы подключения по 3 проводам можно отнести то, что в схеме не предусмотрено торможение скоростью.

Подключение драйвера мотора к Arduino по 4 проводам:

Выводы «L_EN», «R_EN», «L_PWM» и «R_PWM» подключаются к выводам Arduino поддерживающим ШИМ (иначе целесообразнее использовать схему с 2 или 3 проводами).

Управление драйвером мотора по 4 проводам:

В такой схеме можно осуществлять управление по любому из рассмотренных ранее вариантов управления, переключаясь с одной схемы на другую по мере необходимости.

Преимущества и недостатки схемы управления по 4 проводам:

Достоинство схемы заключается в её универсальности, можно использовать достоинства двух предыдущих схем и избавиться от их недостатков.

Явный недостаток схемы - максимальное количество задействованных выводов Arduino.

Питание

- Напряжение питания двигателя (5,5 - 27,5 В постоянного тока) подаётся на выводы «S+» и «S-» клеммника.
- Напряжение питания логической части (5 В постоянного тока) подаётся на выводы «Vcc»

и «GND».

Подробнее о драйвере

Драйвер построен на базе H-моста собранного из двух полумостов на чипах BTS7960. Чипы BTS7960 поддерживают ШИМ до 25 кГц (например, у Arduino UNO частота ШИМ всего 0,5 кГц) и снабжены схемами защиты от короткого замыкания, перегрева, перенапряжения (на выводах клеммника S+ и S-) и падения напряжения ниже 5,5 В (на выводах клеммника S+ и S-). Чипы BTS7960 имеют вывод состояния «IS» напряжение на котором меняется пропорционально току протекающему через двигатель (что позволяет определить, какую нагрузку испытывает двигатель), а в случае обнаружения ошибок, вывод «IS» работает как флаг обнаружения ошибки устанавливаясь в логическую «1». На плате выводы чипов «IS» прижаты к GND через сопротивление 10 кОм и подключены к выходам платы «L_IS» и «R_IS». На плате драйвера установлен шинный формирователь на базе чипа 74HC244 который обеспечивает развязку логических уровней управляющих сигналов между входами «L_EN», «R_EN», «L_PWM», «R_PWM» и входами чипов BTS7960. Благодаря наличию шинного формирователя управлять драйвером можно логическими уровнями как 3,3 В, так и 5 В логики.

Примеры

Управление драйвером мотора

В приведённом скетче используется схема управления драйвером по 3 проводам. Номера выводов Arduino назначаются в трёх первых строках скетча.

```
const uint8_t EN = 3; // № вывода Arduino к которому подключены входы драйвера L_EN и R_EN.
Можно указать любой вывод Arduino поддерживающий ШИМ.
const uint8_t L_PWM = 4; // № вывода Arduino к которому подключён вход драйвера L_PWM.
Можно указать любой вывод Arduino, как цифровой, так и аналоговый.
const uint8_t R_PWM = 5; // № вывода Arduino к которому подключён вход драйвера R_PWM.
Можно указать любой вывод Arduino, как цифровой, так и аналоговый.
void setup(){
pinMode(EN, OUTPUT); // Конфигурируем вывод EN как выход (выход Arduino, вход драйвера)
pinMode(L_PWM, OUTPUT); // Конфигурируем вывод L_PWM как выход (выход Arduino, вход
драйвера)
pinMode(R_PWM, OUTPUT); // Конфигурируем вывод R_PWM как выход (выход Arduino, вход
драйвера)
}
void loop(){
// Движение вперёд на 50% скорости:
digitalWrite(L_PWM, LOW ); // Устанавливаем логический 0 на входе драйвера L_PWM, значит
на выходе драйвера M- будет установлен потенциал S-
digitalWrite(R_PWM, HIGH); // Устанавливаем логическую 1 на входе драйвера R_PWM, значит
на выходе драйвера M+ будет установлен потенциал S+
analogWrite (EN, 127 ); // Устанавливаем 50% ШИМ на входах драйвера L_EN и R_EN, это
скорость, можно установить от 0 (0%) до 255 (100%).
delay(3000); // Ждём 3 секунды. ШИМ и логические уровни останутся без изменений, значит
мотор продолжит вращаться с указанной скоростью и направлением.
// Движение вперёд на 100% скорости:
digitalWrite(L_PWM, LOW ); // Устанавливаем логический 0 на входе драйвера L_PWM, значит
на выходе драйвера M- будет установлен потенциал S-
digitalWrite(R_PWM, HIGH); // Устанавливаем логическую 1 на входе драйвера R_PWM, значит
на выходе драйвера M+ будет установлен потенциал S+
```

```
analogWrite(EN, 255 ); // Устанавливаем 100% ШИМ на входах драйвера L_EN и R_EN, это
скорость. Если устанавливаемое значение 255, то функцию можно заменить на digitalWrite(EN,
HIGH);
delay(3000); // Ждём 3 секунды. ШИМ и логические уровни останутся без изменений, значит
мотор продолжит вращаться с указанной скоростью и направлением.
// Свободное вращение:
digitalWrite(EN, LOW ); // Устанавливаем логический 0 на входах драйвера L_EN и R_EN, значит
выходы М+ и М- перейдут в состояние высокого импеданса и мотор будет электрически
отключён.
delay(3000); // Ждём 3 секунды. Логические уровни на входах драйвера L_PWM и R_PWM не
имеют значения (могут быть любыми).
// Движение назад на 50% скорости:
digitalWrite(L_PWM, HIGH); // Устанавливаем логическую 1 на входе драйвера L_PWM, значит
на выходе драйвера М- будет установлен потенциал S+
digitalWrite(R_PWM, LOW ); // Устанавливаем логический 0 на входе драйвера R_PWM, значит
на выходе драйвера М+ будет установлен потенциал S-
analogWrite(EN, 127 ); // Устанавливаем 50% ШИМ на входах драйвера L_EN и R_EN, это
скорость, можно установить от 0 (0%) до 255 (100%).
delay(3000); // Ждём 3 секунды. ШИМ и логические уровни останутся без изменений, значит
мотор продолжит вращаться с указанной скоростью и направлением.
// Движение назад на 100% скорости:
digitalWrite(L_PWM, HIGH); // Устанавливаем логическую 1 на входе драйвера L_PWM, значит
на выходе драйвера М- будет установлен потенциал S+
digitalWrite(R_PWM, LOW ); // Устанавливаем логический 0 на входе драйвера R_PWM, значит
на выходе драйвера М+ будет установлен потенциал S-
digitalWrite(EN, HIGH); // Эта функция выполнит те же действия что и функция analogWrite(EN,
255);
delay(3000); // Ждём 3 секунды. Логические уровни останутся без изменений, значит мотор
продолжит вращаться с указанной скоростью и направлением.
// Торможение с силой 50%:
digitalWrite(L_PWM, HIGH); // Устанавливаем логическую 1, но можно установить и логический
0, главное что бы уровни на входах драйвера L_PWM и R_PWM совпадали.
digitalWrite(R_PWM, HIGH); // Устанавливаем логическую 1, но можно установить и логический
0, главное что бы уровни на входах драйвера L_PWM и R_PWM совпадали.
analogWrite(EN, 127 ); // Устанавливаем 50% ШИМ на входах драйвера L_EN и R_EN, это сила
торможения, можно установить от 0 (0%) до 255 (100%).
delay(3000); // Ждём 3 секунды. За это время мотор остановится.
}
```