

IX Всероссийская научно-практическая конференция для студентов и учащейся молодежи  
«Прогрессивные технологии и экономика в машиностроении»

должен способствовать снижению издержек налогоплательщиков, связанных с использованием ККТ, а также сократить теневой оборот наличности и, как следствие, увеличить налоговые поступления.»

Все налогоплательщики, которые используют ККТ, должны официально её зарегистрировать и заключить договор с ОФД на обработку фискальных данных.

Законом предусмотрен плавный переход на новую схему работы налогоплательщиков с контролирующим органом (ФНС), включающую оператора фискальных данных, как обязательного.

#### **АСК НДС-2.**

Как стало известно, новость от 22 сентября 2014 года рассказывает, что: «с 1 января 2015 года, сдавая декларации по НДС, начнется с вами загрузка книг покупок и продаж, а также журнала выставленных и принятых счетов-фактур для посредников. Налоговики придумали, что теперь вся информация будет накапливаться в системе «Big data», эту систему готовит ФНС России. Для работы в этой системе создана специальная программа «АСК НДС-2»[4]. ФНС России уже дало ознакомительное время своим специалистам с этой программой. Работа программы состоит в следующем. Все счета-фактур станут попадать в общероссийскую базу. Программа сама проводит сравнение данных о каждой операции по цепочке движения товара. Система покажет инспекторам налоговые разрывы в операциях, по которым есть вычеты, но не уплачен НДС. По таким расхождениям налоговики смогут требовать у компаний счета-фактур, а также первичную документацию. Чем больше та или иная компания в своих цепочках использует «обнальные конторы» [4], тем больше следов в базе будет оставаться. Дальше все это будет высылаться налоговикам и будет сразу понятно, на какие суммы НДС поставлен к вычету или к возмещению по сути дела не законно.

Литература.

1. Государственная информационная система маркировки [Электронный ресурс]. Режим доступа: <https://www.narodifrossii.ru/?p=13914> (Дата обращения: 26.02.2018)
2. ФГИС «Меркурий» [Электронный ресурс]. Режим доступа: <http://xn---8sbmbbmccjipfvkcfubdkla2b8cyk.xn--p1ai/%D0%B3%D0%B8%D1%81-%D0%BC%D0%B5%D1%80%D0%BA%D1%83%D1%80%D0%B8%D0%B9> / Дата обращения: 26.02.2018г.
3. Оператор фискальных данных [Электронный ресурс]. Режим доступа: [https://ru.wikipedia.org/wiki/Оператор\\_фискальных\\_данных](https://ru.wikipedia.org/wiki/Оператор_фискальных_данных). (Дата обращения: 26.02.2018)
4. «АСК НДС-2» – новый сервис ФНС для поиска незаконных вычетов НДС [Электронный ресурс]. Режим доступа: <https://turov.pro/ask-nds-2-novyiy-servis-fns-dlya-poiska-nezakonnyih-vyichetov-nds/>. (Дата обращения: 27.02.2018)

#### **СТАБИЛИЗАЦИЯ ДВИЖЕНИЯ РОБОТА ПО ПОКАЗАНИЯМ ЭЛЕКТРОННОГО КОМПАСА**

*А.Д. Веретенников, А.В. Сосновская студенты группы,  
научный руководитель: Момот М.В.*

*Юргинский технологический институт (филиал) Национального исследовательского  
Томского политехнического университета  
652055, Кемеровская обл., г. Юрга, ул. Ленинградская, 26  
E-mail: [momotmvu@yandex.ru](mailto:momotmvu@yandex.ru)*

Цель – стабилизация движения робота по показаниям электронного компаса.

Компас – это прибор помогающий определить направление на магнитный полюс земли. Магнитный и географический северные полюса отличаются. Северный магнитный полюс сдвинут в район канадской Арктики, и передвигается на 10 километров в год.

Плата электронного компаса НМС5883L изображена на рис. 1.

Подключение электронного компаса к плате Arduino показано на рис.2

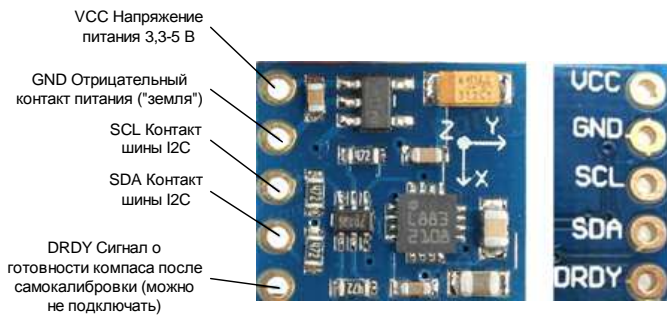


Рис. 1 - Электронный компас HMC5883L

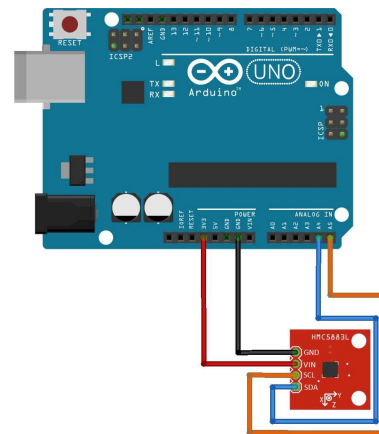


Рис. 2 - Порядок подключения платы HMC5883L к Arduino UNO

Получение данных от электронного компаса

Программа получения данных от HMC5883L (листинг 1).

Листинг 1. Получение данных от электронного компаса

```
//Подключаем библиотеку I2C.
#include <Wire.h>
/////////////////////////////////////////////////////////////////
// адрес i2c компаса.
#define address 0x1E
void setup() {
  // Серийный порт для связи с ПК.
  Serial.begin(9600);
  //Запускаем связь по шине I2C.
  Wire.begin();
  //Для анализа окончания процесса будем использовать светодиод.
  pinMode(6, OUTPUT);
}
//=====
void loop() {
  int x, y, z; //triple axis data
  Wire.beginTransmission(address);
  Wire.write(0x03); //select register 3, X MSB register
  Wire.endTransmission();
  Wire.requestFrom(address, 6);
  if (6 <= Wire.available()) {
    x = Wire.read() << 8;  x |= Wire.read();
    z = Wire.read() << 8;  z |= Wire.read();
    y = Wire.read() << 8;  y |= Wire.read();
  }
  Serial.print("x: "); Serial.print(x);
  Serial.print(" y: "); Serial.print(y);
  Serial.print(" z: "); Serial.println(z);
  delay(500);
}
```

Результаты работы программы представлены на рисунке 3.

Как можно заметить из рисунка 3 данные по разным осям заметно различаются. Электронный компас невозможно использовать без калибровки.

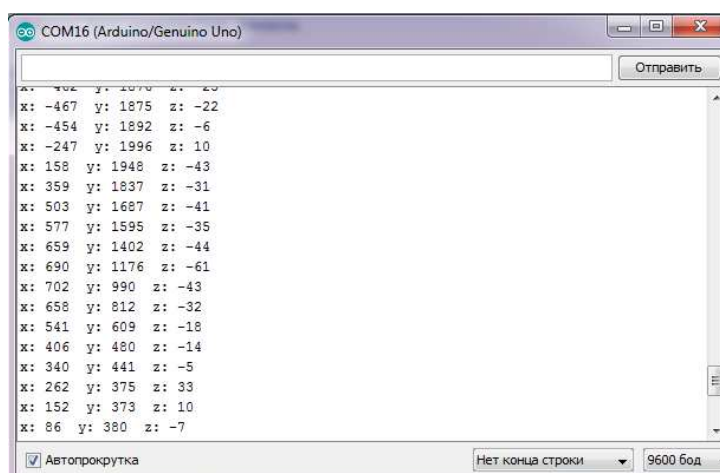


Рис. 3 - Непосредственно полученные данные от электронного компаса  
Для калибровки потребуется вращать робота и измерять его показания, затем, используя полученные экстремумы произвести калибровку осей робота и положения нуля по осям.

Листинг 2. Калибровка компаса

```
//Подключаем библиотеку I2C.
#include <Wire.h>
////////////////////////////////////
//Подключаю функции HMC5883L
#include "compas.h"
//=====
void setup() {
  // Серийный порт для связи с ПК.
  Serial.begin(9600);
  //Запускаем связь по шине I2C.
  Wire.begin();
  //Компас
  compas_setup(); // инициализация HMC5883L
  //Для анализа окончания процесса будем использовать светодиод.
  pinMode(6, OUTPUT);
}
//=====
void loop() {
  //Вращаем робота вокруг своей оси
  //Вклчение светодиода.
  digitalWrite(6, HIGH);
  compas_max_min_to_eeprom();
  //Выклчение светодиода.
  digitalWrite(6, 0);
  delay(1000);
}
```

Файл compas.h содержит функцию калибровки компаса, и сервисные функции, например записи калибровочных коэффициентов в энергонезависимую память.

Листинг 3. Функции файла compas.h

```
#include <EEPROM.h>
#include <Wire.h>
// адрес i2c компаса.
#define address 0x1E
float dX, dY, dZ, scaleX, scaleY;
////////////////////////////////////
```

```

void write_int_to_eeprom(int ny, int data) //Младший адрес и переменная
{
  byte temp = data; // младший байт;
  EEPROM.write(ny, temp);
  temp = data >> 8;
  EEPROM.write(ny + 1, temp);
}
/////////////////////////////////////////////////////////////////
int read_int_from_eeprom(int ny) //Младший адрес и переменная
{
  int temp = EEPROM.read(ny); //младший байт
  int temp2 = EEPROM.read(ny + 1); //старший байт
  return (temp | temp2 << 8);
}
/////////////////////////////////////////////////////////////////
void compas_max_min_to_eeprom()
{
  int minx, miny, minz;
  int maxx, maxy, maxz;
  unsigned long timer;
  timer = millis() + 60000;
  minx = miny = minz = maxx = maxy = maxz = 0;
  do {
    int x, y, z; //triple axis data
    Wire.beginTransmission(address);
    Wire.write(0x03); //select register 3, X MSB register
    Wire.endTransmission();
    Wire.requestFrom(address, 6);
    if (6 <= Wire.available()) {
      x = Wire.read() << 8;  x |= Wire.read();
      z = Wire.read() << 8;  z |= Wire.read();
      y = Wire.read() << 8;  y |= Wire.read();

      if (x < minx) minx = x;
      if (y < miny) miny = y;
      if (z < minz) minz = z;

      if (x > maxx) maxx = x;
      if (y > maxy) maxy = y;
      if (z > maxz) maxz = z;
    }
    Serial.print("x: "); Serial.print(x);
    Serial.print(" y: "); Serial.print(y);
    Serial.print(" z: "); Serial.println(z);
    delay(1);
  } while (millis() < timer); // В течении минуты крутим робота
  // Записываем значения в EEPROM
  write_int_to_eeprom(0, minx);
  write_int_to_eeprom(2, maxx);
  write_int_to_eeprom(4, miny);
  write_int_to_eeprom(6, maxy);
  write_int_to_eeprom(8, minz);
  write_int_to_eeprom(10, maxz);
}

```

```
////////////////////////////////////  
////////////////////////////////////  
void compas_setup() {  
  Wire.begin();  
  Wire.beginTransmission(address);  
  Wire.write(0x02);  
  Wire.write(0x00);  
  Wire.endTransmission();  
  // Рассчитываем коррективы по данным их еепром.  
  dX = float(read_int_from_eeprom(0) + read_int_from_eeprom(2)) / 2.0;  
  dY = float(read_int_from_eeprom(4) + read_int_from_eeprom(6)) / 2.0;  
  dZ = float(read_int_from_eeprom(8) + read_int_from_eeprom(10)) / 2.0;  
  scaleX = 1.0 / float(read_int_from_eeprom(2) - read_int_from_eeprom(0));  
  scaleY = 1.0 / float(read_int_from_eeprom(6) - read_int_from_eeprom(4));  
}  
////////////////////////////////////  
float get_Compass_ang() {  
  int x, y, z; //triple axis data  
  Wire.beginTransmission(address);  
  Wire.write(0x03); //select register 3, X MSB register  
  Wire.endTransmission();  
  Wire.requestFrom(address, 6);  
  if (6 <= Wire.available()) {  
    x = Wire.read() << 8;  x |= Wire.read();  
    z = Wire.read() << 8;  z |= Wire.read();  
    y = Wire.read() << 8;  y |= Wire.read();  
    return atan2( (float(y) - dY) * scaleY, (float(x) - dX) * scaleX ) * RAD_TO_DEG;  
  }  
}
```

Следующий листинг уже показывает отклонение угла поворота компаса от магнитного севера.  
Листинг 4. Программа расчета направления на магнитный север с учетом калибровки прибора

```
//Подключаем библиотеку I2C.  
#include <Wire.h>  
////////////////////////////////////  
//Подключаю функции HMC5883L  
#include "compas.h"  
//=====  
void setup() {  
  // Серийный порт для связи с ПК.  
  Serial.begin(9600);  
  //Запускаем связь по шине I2C.  
  Wire.begin();  
  //Компас  
  compas_setup(); // инициализация HMC5883L  
  //Для анализа окончания процесса будем использовать светодиод.  
  pinMode(6, OUTPUT);  
  //=====  
void loop() {  
  float heading = get_Compass_ang();  
  Serial.println(heading);  
  digitalWrite(6, HIGH);  
  delay(500);  
  digitalWrite(6, 0);
```

Рис. 4 демонстрирует работу программы для измерения угла от -180° до 180°,

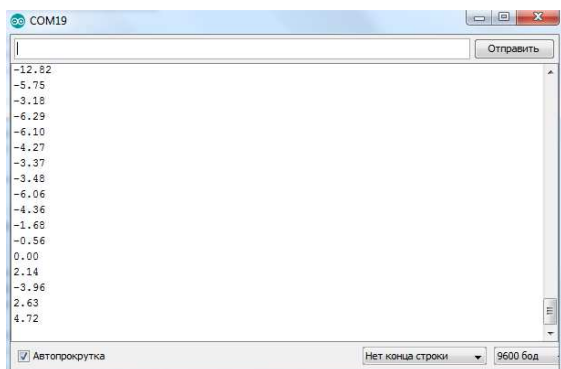


Рис. 4. Результат работы откалиброванного компаса

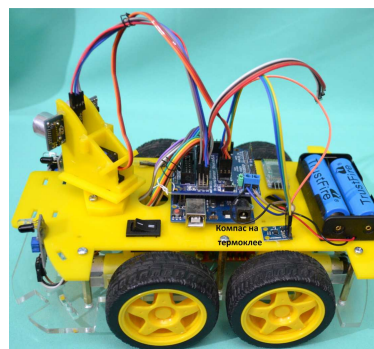


Рис. 5. Вариант установки на робота платы магнитометра HMC5883L

Установка электронного компаса на робота

При установке на робота компас должен быть точно центрирован. Для правильной его ориентации нужно знать направление на северный магнитный полюс. Получить его можно с помощью классического компаса, или компаса в смартфоне.

Программа

Теперь можно приступить к разработке программы движения робота. На рис.6 приведен алгоритм, отвечающий за движение робота в направлении на север, — робот ищет северный магнитный полюс и разворачивается к нему.

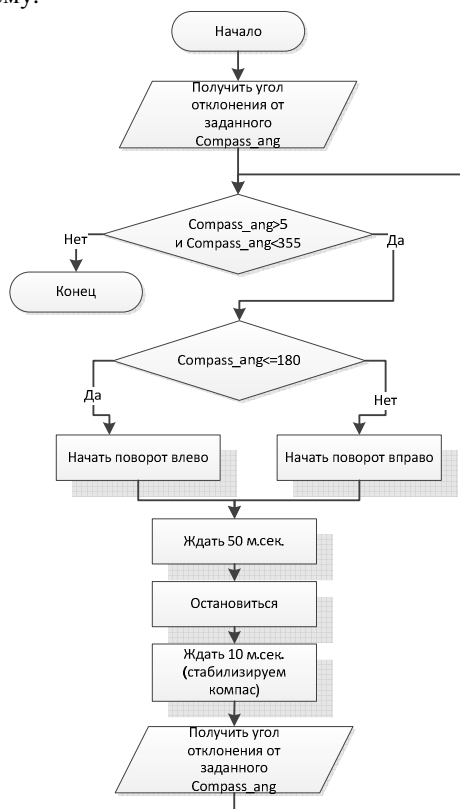


Рис. 6. Алгоритм поворота робота на северный магнитный полюс

Литература.

1. Момот М. В. Мобильные роботы на базе Arduino. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2018. — 336 с.: ил. — (Электроника) ISBN 978-5-9775-3861-9