

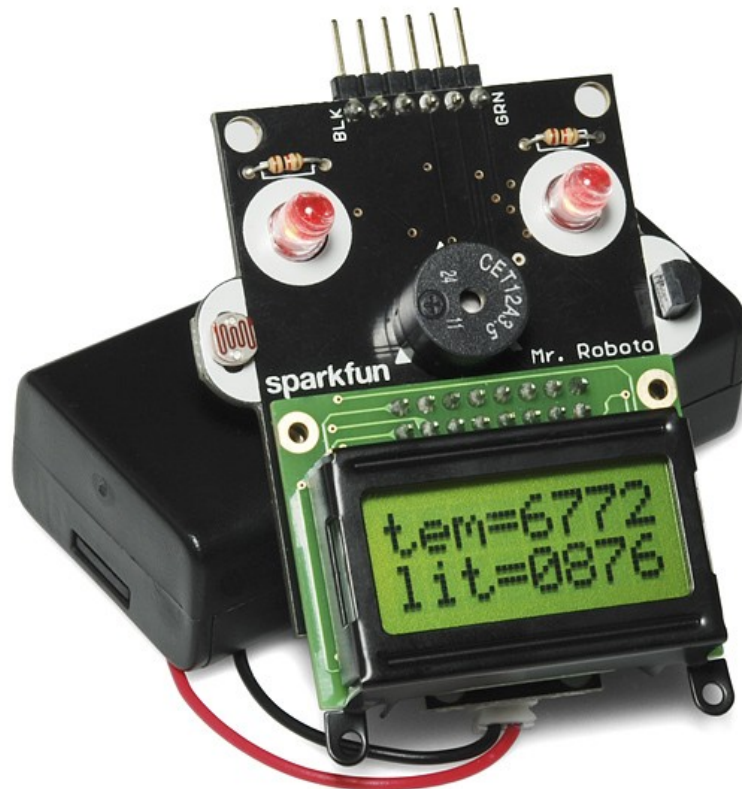
SPARKFUN

Mr. Roboto Kit



Описание: Mr. Roboto Arduino – совместимая отладочная плата, разработанная для обучения программирования в среде Arduino IDE. Все планарные компоненты уже установлены заранее, так что вам остается потратить несколько минут на монтаж штырьковых элементов, после чего плата будет полностью готова к использованию. Совокупность сенсоров, зуммера и ЖК дисплея – является прекрасной платформой для получения навыков программирования микроконтроллерных устройств.

Для программирования Mr. Roboto используется виртуальный COM – порт, на основе микросхемы FT232. Необходимо просто подключить к шестиконтактному разъему данную плату. Далее подключить устройство к USB порту и установить драйвера микросхемы FT232. После чего, остается лишь указать в среде программирования Arduino IDE тип программируемой платы - "Arduino Pro or Pro Mini (5V, 16MHz) w/ ATmega328" и номер порта.



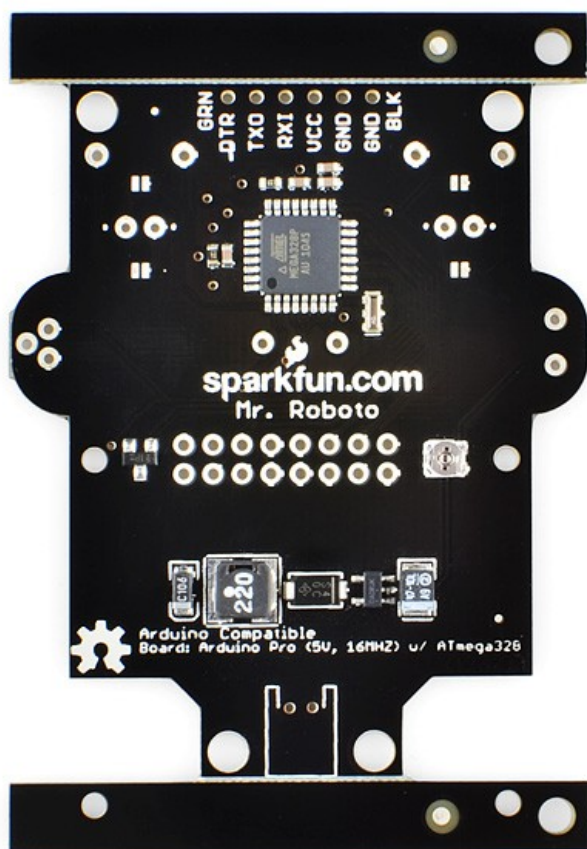
Особенности прибора:

- АТМega328 работает от внешнего кварца на частоте 16МГц, полностью совместимо с Arduino
- Наличие на плате светодиодов и температурных сенсоров
- Пьезоэлемент (зуммер) и ЖК дисплей
- Питание от двух батареек типа АА
- Компоненты поверхностного монтажа (smd) уже установлены

Состав набора:

- Плата Mr. Roboto с уже установленными планарными компонентами
- 1 x миниатюрный ЖК индикатор
- 1 x шестиконтактный разъем
- 1 x шестнадцатиконтактный разъем
- 2 x сверх ярких красных светодиода. Описание:
<http://www.sparkfun.com/datasheets/Components/YSL-R531R3C-A13.pdf>
- 2 x 1кОм резистора 1/6 Вт, +/- 5%
- 1 x миниатюрный фоторезистор. Описание:
<http://www.sparkfun.com/datasheets/Sensors/Imaging/SEN-09088-datasheet.pdf>
- 1 x температурный сенсор TMP36. Описание:
http://dlmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Temp/TMP35_36_37.pdf
- 1 x Зуммер. Описание: <http://www.sparkfun.com/datasheets/Components/CEM-1203.pdf>
- 1 x двухконтактный JST разъем. Описание
<http://www.sparkfun.com/datasheets/Prototyping/Connectors/JST-Horizontal.pdf>

- 2 х батарейки типа АА
- 1 х батарейный отсек с выключателем. Описание: <https://www.sparkfun.com/products/9925?>



Характеристики фоторезистора:

- Световое сопротивление: ~1 кОм
- Темновое сопротивление: ~10 кОм
- Максимальное напряжение: 150 В
- Максимальная рассеиваемая мощность: 100 мВт

Размеры фоторезистора:

- 2 x 4 x 5 мм
- Расстояние между ножками – 4 мм
- Длина ножек 31 мм

Характеристики температурного сенсора:

- входное напряжение: от 2.7 до 5.5 Вольт
- чувствительность - 10 мВ/°С
- точность ±2°С на всем температурном диапазоне сенсора
- линейность ±0.5°С
- диапазон рабочих температур: от -40°С до +125°С

Среда программирования:

В настоящее время Arduino IDE доступно для Windows, Mac OS X и Linux 32bit. Среду можно скачать по ссылке: <http://arduino.cc/en/Main/Software>

Данная программа распространяется в виде архива и не нуждается в установке.

Перед началом программирования необходимо выбрать номер последовательного порта (виртуальный COM – порт) и тип платы. Сделать это можно во вкладке **Service**. В меню **File** -> **Examples** находятся базовые примеры работы с портами Arduino и его библиотеками.

Структура программы:

- Программа начинается с включения всех необходимых библиотек.
- Функция **setup()** необходима для настроек устройства во время запуска. С нее начинается выполнение любой программы, поэтому она, обычно, используется для того, что бы настроить работу микроконтроллера должным образом: конфигурирование прерываний и портов, необходимых для данной задачи. По умолчанию, все порты находятся в режиме входов, что снижает вероятность случайного короткого замыкания.
- Функция **loop()** является основой программы. Ее содержимое будет постоянно повторяться по кругу, пока устройство не будет выключено или перезагружено.

Информация о языке программирования может быть найдена тут:

<http://arduino.cc/en/Reference/HomePage> (на английском языке)

<http://arduino.ru/Reference>

Пример:

```
// Присвоение ножкам уникальных имен
```

```
// Используется для большей простоты запоминания и понимания
```

```
const int leftEyePin = 3; // левый глаз - 3 ножка
```

```
const int rightEyePin = 9; // правый глаз – 9 ножка
```

```
void setup()
```

```
{
```

```
// Перед использованием порта, надо указать его назначение (цифровой вход, либо
```

```
// цифровой выход). В данном случае мы выводим сигнал на светодиоды, поэтому
```

```
// порты надо сконфигурировать как выходы.
```

```
// Для этого напишем:
```

```
pinMode(leftEyePin, OUTPUT);
```

```
pinMode(rightEyePin, OUTPUT);
```

```
// При запуске программы мы хотим, что бы оба светодиода были выключены.
```

```
// Для этого можно использовать функцию digitalWrite()
```

```
// Придавая порту значение LOW - мы выключаем светодиод. В случае значения HIGH –
```

```
// светодиод включится.
```

```
digitalWrite(leftEyePin, LOW);
```

```
digitalWrite(rightEyePin, LOW);
```

```
}
```

```
void loop()
```

```
{
```

```
// Здесь используются функции, которые будут описаны ниже.
```

```
// blinkRightEye() и blinkLeftEye вызываются сначала.  
blinkRightEye(500); // моргнуть правым глазом на 500мс  
blinkLeftEye(500); // моргнуть левым глазом на 500мс
```

```
// Затем будем моргать обоими глазами, используя analogBlinkRight и LeftEye()  
analogBlinkRightEye(1000);  
analogBlinkLeftEye(1000);  
// В конце функции loop() мы снова переходим к ее началу. И так будет продолжаться  
// бесконечно долго, насколько хватит заряда батареи.  
}
```

```
/* blinkRightEye(int blinkTime) – Эта функция включит правый светодиодный глаз Mr. Robot на  
время blinkTime/2 и выключит на время blinkTime/2. В итоге выполнение этой функции займет  
blinkTime миллисекунд.
```

```
*/
```

```
void blinkRightEye(int blinkTime)  
{  
    digitalWrite(rightEyePin, HIGH); // Включаем правый светодиод  
    delay(blinkTime/2); // Ничего не делаем в течении blinkTime/2 миллисекунд  
    digitalWrite(rightEyePin, LOW); // Выключаем правый светодиод  
    delay(blinkTime/2); // Снова ничего не делаем в течении blinkTime/2 миллисекунд  
}
```

```
/* blinkLeftEye(int blinkTime) – Эта функция включит левый светодиодный глаз Mr. Robot на  
время blinkTime/2 миллисекунд и выключит через время blinkTime/2 миллисекунд. В итоге та  
функция потребует blinkTime миллисекунд на выполнение.
```

```
*/
```

```
void blinkLeftEye(int blinkTime)  
{  
    digitalWrite(leftEyePin, HIGH); // Включаем левый светодиод  
    delay(blinkTime/2); // Ничего не делаем в течении blinkTime/2 миллисекунд  
    digitalWrite(leftEyePin, LOW); // Выключаем левый светодиод  
    delay(blinkTime/2); // Ничего не делаем в течении blinkTime/2 миллисекунд  
}
```

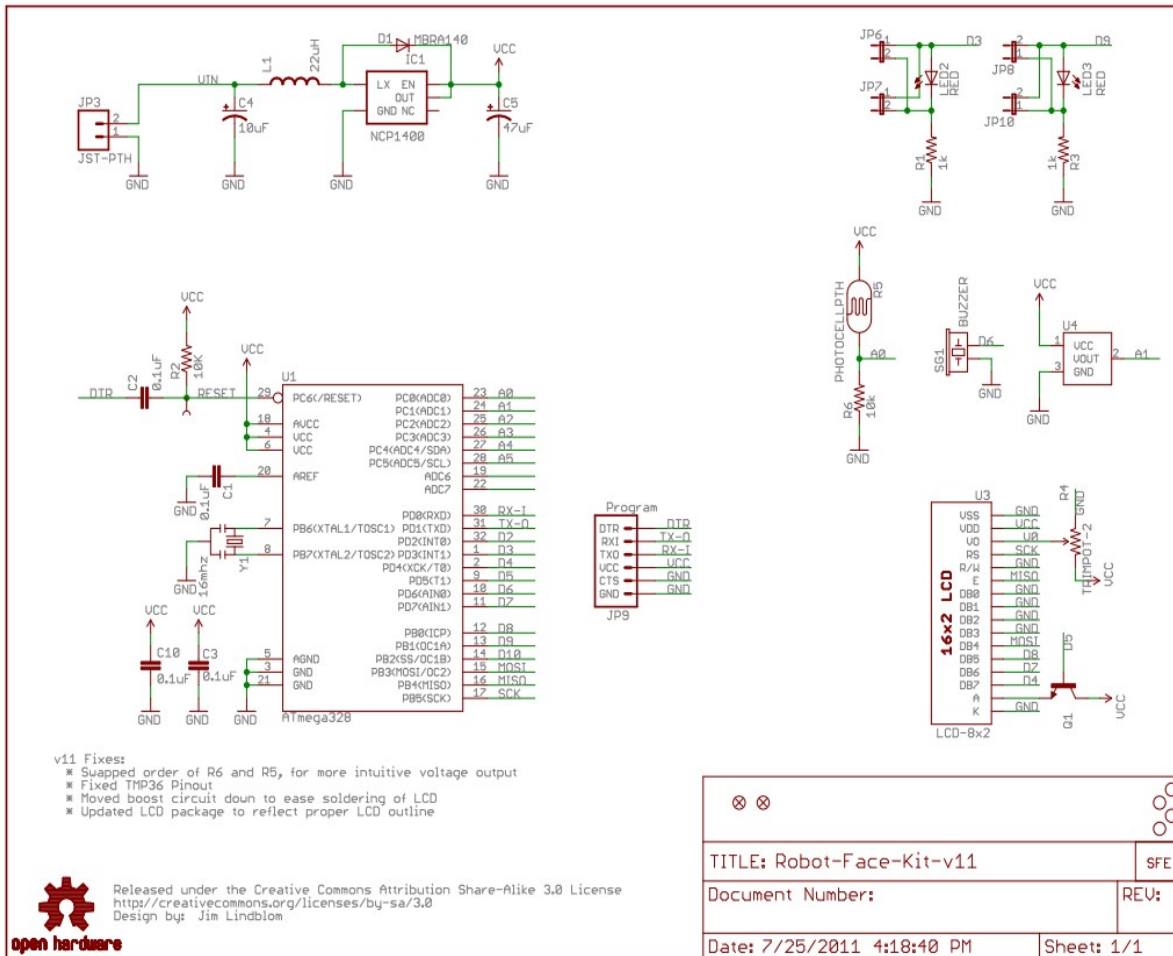
```
/* analogBlinkLeftEye(int blinkTime) – Эта функция использует  
analogWrite() вместо digitalWrite() что бы включить левый светодиодный глаз и постепенно  
выключить его*/
```

```
void analogBlinkLeftEye(int blinkTime)  
{  
    // мы собираемся разделить каждую паузу на  
    // на маленькие интервалы от blinkTime.  
    // Далее, нам понадобится эта переменная.  
    long blinkIntervalMicroseconds = ((long)blinkTime*1000/512);  
  
    for (int i=0; i<=255; i++)  
    {  
        // analogWrite() отличается от digitalWrite, тем, что мы можем задать значения не только  
        // HIGH или LOW, но и различные значения внутри данного диапазона: от 0 (LOW) до //255  
        // (HIGH).  
        analogWrite(leftEyePin, i);  
        delayMicroseconds(blinkIntervalMicroseconds);  
    }  
}
```

```
}  
for (int i=255; i>=0; i--)  
{  
  analogWrite(leftEyePin, i);  
  delayMicroseconds(blinkIntervalMicroseconds);  
}  
}  
  
void analogBlinkRightEye(int blinkTime)  
{  
  long blinkIntervalMicroseconds = ((long)blinkTime*1000/512);  
  
  for (int i=0; i<=255; i++)  
  {  
    analogWrite(rightEyePin, i);  
    delayMicroseconds(blinkIntervalMicroseconds);  
  }  
  
  for (int i=255; i>=0; i--)  
  {  
    analogWrite(rightEyePin, i);  
    delayMicroseconds(blinkIntervalMicroseconds);  
  }  
}
```

Эта и другие программы для Mr. Roboto доступны на сайте производителя:
<http://dlnmh9ip6v2uc.cloudfront.net/datasheets/Kits/MrRoboto.zip>

Принципиальная схема устройства:



Released under the Creative Commons Attribution Share-Alike 3.0 License
<http://creativecommons.org/licenses/by-sa/3.0>
 Design by: Jim Lindblom

TITLE: Robot-Face-Kit-v11		SFE
Document Number:		REV:
Date: 7/25/2011 4:18:40 PM	Sheet: 1/1	