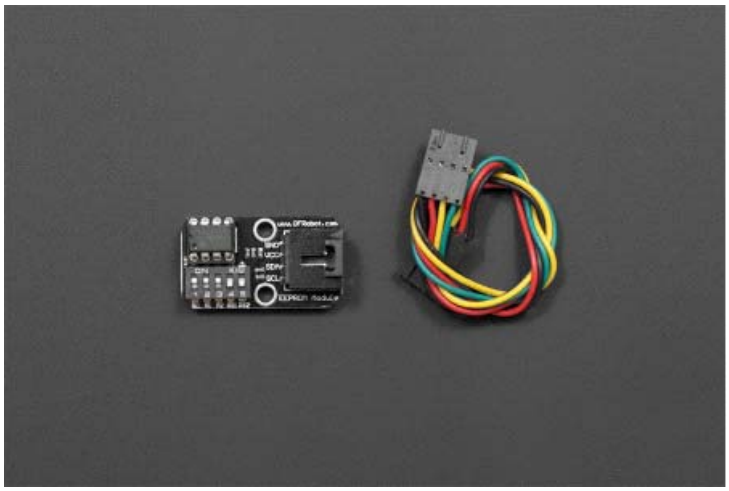
EEPROM Data Storage Module For Arduino (SKU:DFR0117)



Contents

Introduction

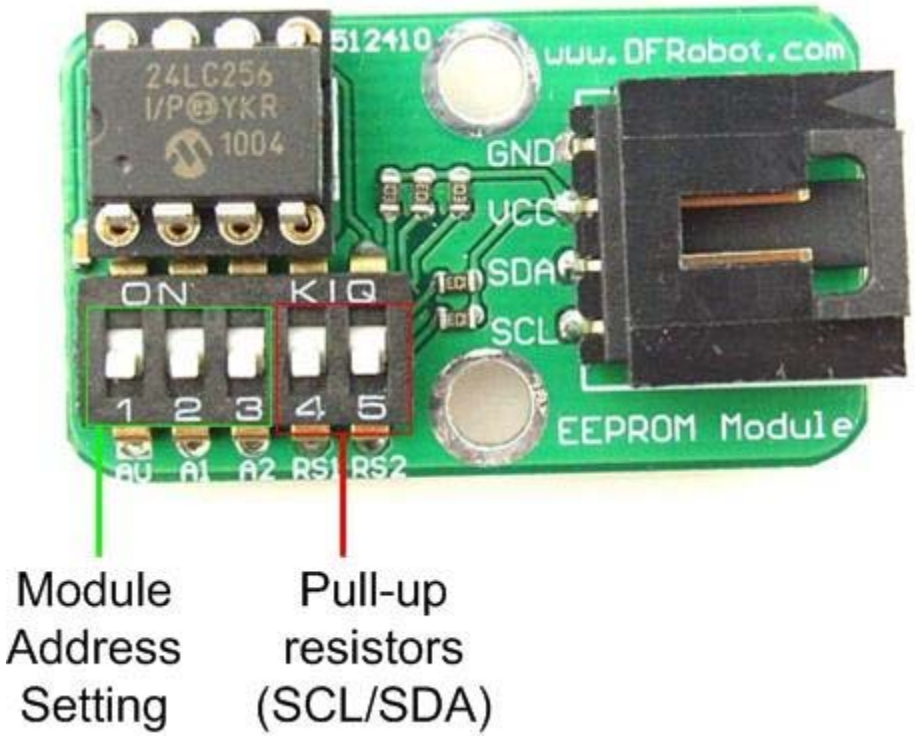
If you want more storage for your Arduino project, this module is designed for you. This EEPROM data storage module supplies an extra 32k byte for your project. Simply wired them via I2C. Supplied with Interface cable which can be fawlessly integrated with interface shield.

Specification

- Supply Voltage:+5V
- Current:<10mA
- Interface:I2C/TWI
- EEPROM: AT24C256
- Weight:5 gram

Setting

- Default Address:0x50
- Adjustable from 0x50-0x57 via the swither



Address Truth Table

A0	A1	A2	Add
ON	ON	ON	0x50
OFF	ON	ON	0x51
ON	OFF	ON	0x52
OFF	OFF	ON	0x53
ON	ON	OFF	0x54
OFF	ON	OFF	0x55
ON	OFF	OFF	0x56
OFF	OFF	OFF	0x57

Sample Code

```
/*
  *  Use the I2C bus with EEPROM 24LC64
  *  Sketch:    eeprom.pde
  *
  *  Author: hkhijhe
  *  Date: 01/10/2010
  *
  *
  */


  #include <Wire.h> //I2C library




  void i2c_eeprom_write_byte( int deviceaddress, unsigned int eeaddress, byte
data ) {
    int rdata = data;
    Wire.beginTransmission(deviceaddress);
    Wire.send((int)(eeaddress >> 8)); // MSB
    Wire.send((int)(eeaddress & 0xFF)); // LSB
    Wire.send(rdata);
    Wire.endTransmission();
  }


  // WARNING: address is a page address, 6-bit end will wrap around
  // also, data can be maximum of about 30 bytes, because the Wire library ha
s a buffer of 32 bytes
  void i2c_eeprom_write_page( int deviceaddress, unsigned int eeaddresspage,
byte* data, byte length ) {
    Wire.beginTransmission(deviceaddress);
    Wire.send((int)(eeaddresspage >> 8)); // MSB
    Wire.send((int)(eeaddresspage & 0xFF)); // LSB
    byte c;
    for ( c = 0; c < length; c++)
```

```
      Wire.send(data[c]);
    Wire.endTransmission();
  }


  byte i2c_eeprom_read_byte( int deviceaddress, unsigned int eeaddress ) {
    byte rdata = 0xFF;
    Wire.beginTransmission(deviceaddress);
    Wire.send((int)(eeaddress >> 8)); // MSB
    Wire.send((int)(eeaddress & 0xFF)); // LSB
    Wire.endTransmission();
    Wire.requestFrom(deviceaddress,1);
    if (Wire.available()) rdata = Wire.receive();
    return rdata;
  }


  // maybe let's not read more than 30 or 32 bytes at a time!
  void i2c_eeprom_read_buffer( int deviceaddress, unsigned int eeaddress, byt
e *buffer, int length ) {
    Wire.beginTransmission(deviceaddress);
    Wire.send((int)(eeaddress >> 8)); // MSB
    Wire.send((int)(eeaddress & 0xFF)); // LSB
    Wire.endTransmission();
    Wire.requestFrom(deviceaddress,length);
    int c = 0;
    for ( c = 0; c < length; c++ )
      if (Wire.available()) buffer[c] = Wire.receive();
  }




  void setup()
  {
    char somedata[] = "this is data from the eeprom"; // data to write
```

```
    Wire.begin(); // initialise the connection
    Serial.begin(9600);
    i2c_eeprom_write_page(0x50, 0, (byte *)somedata, sizeof(somedata)); // wr
ite to EEPROM


    delay(10); //add a small delay


    Serial.println("Memory written");
  }


  void loop()
  {
    int addr=0; //first address
    byte b = i2c_eeprom_read_byte(0x50, 0); // access the first address from
the memory


    while (b!=0)
    {
      Serial.print((char)b); //print content to serial port
      addr++; //increase address
      b = i2c_eeprom_read_byte(0x50, addr); //access an address from the memo
ry
    }
    Serial.println(" ");
    delay(2000);


}
```