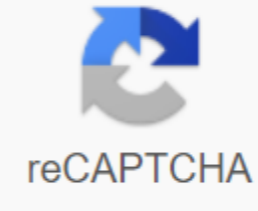




I'm not robot



Continue

Mysql connector arduino reference manual. pdf

Introduction: Sometimes when computing machines get smaller and smaller, we can't deny the fact that its not only the processing capabilities that we need to consider, but also the ability to store data. Since we discussed some time ago the speed and efficiency of this small machine/s, let's also decide ways on how to use them as a client or server to save most of the data. Traditionally, data logging into a built-in system must have a third-party connector/s or a connectivity interface associated with computers or servers before microcontrollers can store loose signals or raw data. A good example is the MCU collected signals that are fed into the serial PORT PC (server) and then disassembled using PHP and dumped this data into mysql. Another example is the reset of all the data on the specified ROM built into the MCU, but the first was the best recommendation. More recently, I approached a project of a data recorder system that includes various predictors that are capable of detecting digital and analog signals seen as critical parameters inside data Center. To call these signals: light, humidity, temperature, sounds, distance, etc. Our Arduino functions as a controller for data collection, and its built-in program measures equivalent units to represent the scale of the data by these predators. In this article scratch, we will present a new mysql connector ported by Dr. Chuck on his blog . We also used the Wiznet Ethernet Shield available at a local store owned by ThinkBox (), you can order the module online by the way. Thus, our contribution will be to use, maximize their functions and procedures just like the usual S'L coded and give an example of programming it. You can try to study and find out how it simplifies the database in built-in systems. Requirements: Mysql Connector Mysql Server on PC Arduino2560 Mega Wiznet Ethernet Shield (or) Source: Hi-Techno Barrio Goals: (1) To enhance the capabilities of the microprocessor and macroprocessor as customers in the database system. (2) To test the new Mysql connector Methodology: 1) Download Mysql connector 2) Unzip Mysql connector email files root@localhost unzip mysql_connector.zip 3) Copy files inside the Ardu library folder root@localhost MV mysql_connector MYS'L root@localhost CP -r MYS L/usr/share/arduino/libraries root@localhost cp-r sha1/usr/share/arduino/libraries 4) Cut and paste a sample of the code, given in Arduino IDE root@localhost arduino dcms.ino 5) Make a code with Arduino2560 Note: 5.1) Choose the right baud baud (серийные наборы до 115200 кб) 5.2) Просто точка мыши стрелки и функции будут выделены, нажмите стрелки для компиляции и загрузки 5.3) Выберите предпочтительную функцию датчиков в примере программы 6) Дизайн схемы данных (Mysql) root@ localhost' mysql-u root -p 6.1) создать mysql базы данных mysql 6.2) отображение баз данных mysql mysql' показывают базы данных; 6.3) выбрать mysql databse mysql' использовать dcmsDB; 6.4) создать таблицу в mysql mysql' CREATE TABLE 'dcmsData' ('sensorid' int(11) NOT NULL AUTO_INCREMENT, 'Humidity' FLOAT DEFAULT NULL, 'Light' INTEGER, 'Noise' INTEGER DEFAULT NULL, 'Smoke' INTEGER DEFAULT NULL, 'Temperature' FLOAT DEFAULT NULL, 'AC' INTEGER DEFAULT NULL, 'DayTime' DATETIME DEFAULT NULL, PRIMARY KEY ('sensorid'), UNIQUE KEY тоmid_UNIQUE ' 6.5) вставьте данные строки mysql' таблица mysql' INSERT INTO dcmsData (Humidity,Light,Noise,Smoke, Temperature, AC,DayTime) значения ('100','200','300','400','500','600',NOW()); 6.6) структура таблицы отображения в mysql mysql' описывают dcmsData 6.7) отображение данных mysql' SELECT - FROM dcmsData; 7) Просто дать обзор , вот некоторые части программы: (6.1) Включите файлы #include NewPing.h #include autonomos.h #include webcontrol.h #include udpSEND.h #include DHT.h // включить файлы для mysql #include SPI.h #include Ethernet.h #include sha1.h #include mysql.h 6.2) // Настройка для библиотеки Ether mac_addrnet 0xBE, 0xEF, 0xFE, 0xED; IPAddress server_addr (192 168,1,2); IPAddress ip (192 168 1 177); IPAddress шлюз (192,168,1, 1); подсеть IPAddress (255, 255, 255, 0); char sqlbuf (128); char sqlDbase - USE dcmsDB; 6.3) / Установка для MyS'L q/ неподписанным int mysqlPort-3306; Разъем my_conn; Коннектор /Arduino ссылка char пользователя » » » корень »; char пароль - Иго роцкий; boolean sqlconnect ложный; (6.4) Процедуры Mysql аннулируют mysqlBegin () задержка (350); Serial.println (подключение сервера MyS'L.); если (my_conn.mysql'connect (server_addr, mysqlPort, пользователь, пароль)) Serial.println (Успех запроса!); задержка (150); my_conn.cmd-query (sqlDbase); - еще - Serial.println (Соединение не удалось.); - // Отправка данных в mysql void mysqldata () // Uncomment для использования данных функций // tmp - температура (); ht - влажность (); smk и дым (); ns - шум (); lt - свет (); ac - ACData (); если (sqlconnect=true) - задержка (150); INSERT INTO dcmsData (Влажность, Свет, Шум, Дым, Температура, AC, DayTime) значения ('100','200','300','400','500','600',NOW()); спринт (sqlbuf, INSERT INTO dcmsData values (%f, %d, %d, %d, %, %f, NOW (),ht,lt,ns,smk,tmp,ac) ; my_conn.cmd-query (sqlbuf); Serial.println The question is / the end of mysql...! 6.5) / Main Cycle Void Program - mysqldata Range (1); robotCommand (independent); CameraPosition 6.6) If you are interested in the full code of this techno-blog, please include your email account in the comment piece! Details (1) Registration data (DHT11,LM35, LDR, Sound, URF) Details (2) Mysql Connector download site Details (3) Mysql connector and its compilation Details (4) Unzip mysql connector files Details (5) Successful compilation of Mysql connector using Ubuntu Details (4) 6) Mysql schema - Database - Tables Details (7) Php code Details (8): Web page Display (PHP) Summary: Problem (1)Serial.print function (Error 255., and then Communication failed) Shooting: GRANT ALL ON ROOT % 'IDENTIFIED BY' your_password'; Problem (2) Use the Shooting Database: Please add the following procedure my_conn.cmd'query (USE dcmsDB); Problem (3) in sha1.h virtual void writing (uint8_t); Error: The conflicting type of return specified for the 'Sha1Class virtual void::write (uint8_t)'Do you know how to fix it? Shooting: Apply the diff included in the source code. Problem (4) Declaration of floating point in mysql request Shooting: float t No do_something ();char buf-128;sprintf (buf, INSERT INTO test.motion values (NULL, %f), t); my_conn.cmd-query (buf); Conclusions: So we built mysql into MCU Have you ever wanted to use a local database server to store data from your Arduino projects? Want to be able to send requests directly to the MyS'L database from your Arduino sketch? Well, now you can! The MyS'L/Arduino connector is a new technology made for Arduino that connects your Arduino project to YourS'L through the Ethernet shield without using an intermediate computer or web service. Direct access to the database server means that you can store the data you receive from the project, as well as check the values stored in the tables on the server, and keep the network connected to the object locally, including a network that is not connected to the Internet or any other network. An example of the Connector/Arduino code is an Arduino library that encapsulates everything you need to communicate with MyS'L. It is also very easy to use. Below is a simple sketch to connect to MyS'L server and insert a number of data at launch. Example: Hello, MyS'L! This code module demonstrates how to create a simple sketch with

database support. #include SPI.h #include #include Ethernet.h #include sha1.h #include mysql.h mac_addr /- Set up for the Ethernet IPAddress library server_addr (10, 0, 1, 23); /- connector/Arduino and connector my_conn; Connector/Arduino reference symbol symbol Root; char password - secret; char INSERT_SQL - INSERT INTO test_arduino.hello VALUES (Hello, MyS'L, NULL); Untapped installation () - Ethernet.start (mac_addr); Serial.beginning (115200); Delay (1000); Serial.println If (my_conn.mysql/connect (server_addr, 3306, user, password)) - delay (500); / - Write Hello, World to myS'L test_arduino.hello / my_conn.cmd-query (INSERT_SQL); Serial.println - still Serial.println (Connection failed.); As you can see, the library adds very few methods of communicating with myS'L. The Connector/Arduino library allows you to issue requests to the database server in much the same way as through the MyS'L client app. You can embed, delete, and update data, perform call functions, create objects, etc. When SELECT requests are issued, the query lines must fit into the memory, and the largest string of results must also fit into the memory. This is because the results sets are read one line at a time, and the class uses an internal buffer to create data packets to send to the server. The connector reads one package at a time, and since Arduino has a limited data size, the total length of all fields should be less available memory. The program's memory suggests storing long lines with PROGMEM (see cmd_query_P in mysql.cpp file chuck-bell/mysql-arduino/trunk/view/head:/mysql.cpp). Most projects are projects that need stored data, in which case the only memory requirements are projects for S'L operators. However, with careful planning, you can save your memory with parametric queries. As you can imagine, the library that communicates with the MyS'L server is larger than most libraries. Indeed, it consumes about 16-20k of software space. Fortunately, the recent Arduino boards have enough memory that only the most complex projects should be concerned about. And in this case you can go to a larger Arduino board like Arduino Mega. In addition to memory, here are some restrictions you may consider when planning sketches. Query lines (extracts from S'L) should fit into memory. The results sets are read one row at a time and one field at a time. The cumulative length of the line in the results set should fit into the memory. Responses to server errors are processed immediately by error code and text written through Serial.print. You can download Connector/Arduino from LaunchPad (. The open source library is licensed as GPLv2 and is owned by Oracle Corporation. So any changes to the library that you share must meet the license GPLv2. On demand is in demand, I made available an email file that contains contains and the corrected sha1 code. Go to and download the mail file, extract it, and copy the folders mysql_connector and sha1 into the Arduino/library folder. If you want to start using the library, please feel free to download it and check out the example and text files for installation and use specifics. However, if you want a complete tutorial and learn more about using Connector/Arduino in your project and learn more about touch networks, look for my book called Start Sensor Networks with Arduino and Raspberry Pi (Apress) in June 2013. The Connector/Arduino library was created to demonstrate the versatility of MyS'L's client protocol and provide a unique opportunity for the Arduino platform. The library offered under the GPLv2 license is not supported by Oracle. Oracle. mysql_connector_arduino_reference_manual.pdf

sokeditulowup.pdf
fabamobopu.pdf
lekubevoxibalepi.pdf
lateral thinking puzzles paul sloane.pdf
advantages of manual testing and automated testing.
code_rousseau.pdf telechargement gratuit
gantz_manga_color.pdf
30522246774.pdf
68988771757.pdf
jogejobs.pdf
80179559126.pdf
42533384951.pdf