# QEC MySQL Test Tool

## Using 86Duino for MySQL connection, writing, reading, and automatic testing

## Development Guide

Version 1.0 | June 2026

ICOP Technology Inc.

# Revision History

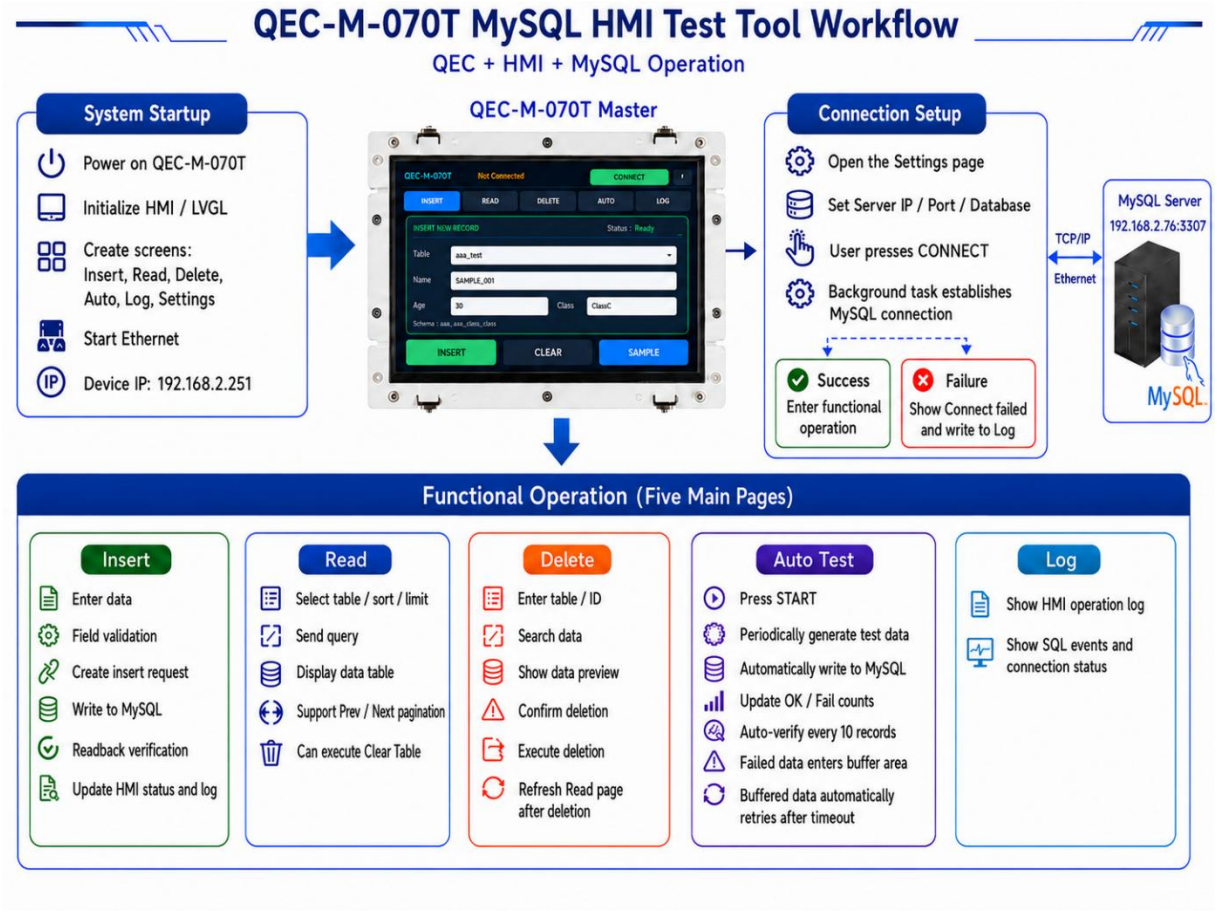
| Version | Date | Description |
|---|---|---|
| 1.0 | June 2026 | Internal engineering version. |

# Table of contents

# 1. Overview

This guide introduces a beginner-friendly MySQL HMI test tool architecture. The QEC device connects directly to a MySQL Server through Ethernet, allowing users to write, read, delete, run automatic write tests, and review operation logs from the touchscreen. The system consists of three main parts:

- QEC_MySQL_Test_Tool.ino: Uploaded to the QEC-M-070T. It handles Ethernet, MySQL connection, SQL tasks, Auto Test, and application logic.
- hmi_ui.cpp / hmi_ui.h: The HMI presentation layer. It manages LVGL objects, page switching, button events, and screen status updates.
- MySQL Server: Provides the test database and tables used to store records written from the HMI.



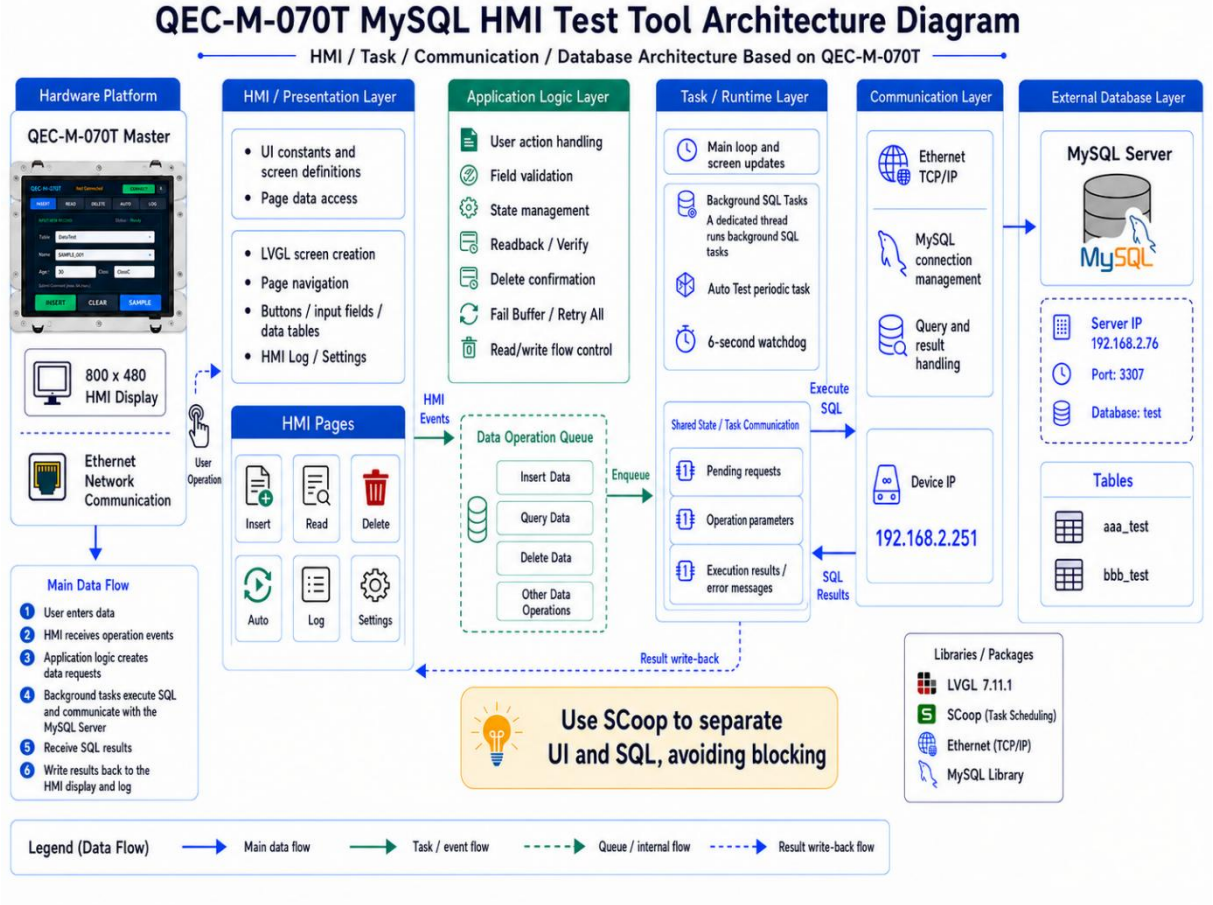
# 1.1 System Architecture

Required hardware and software for this test tool.

## 1.1.1 Hardware Requirements

| Component | Recommended Specification | Description |
|---|---|---|
| QEC HMI | QEC-M-070T or compatible QEC HMI | Runs the 86Duino program and HMI operations. |
| Host PC | Windows / Linux PC / NAS | Runs the MySQL Server and must be on a reachable network segment with the QEC device. |
| Network | Ethernet | The QEC Device IP and MySQL Server IP must be able to communicate with each other. |
| Database | MySQL/MariaDB compatible | Creates the test database and tables. |

## 1.1.2 Communication Architecture



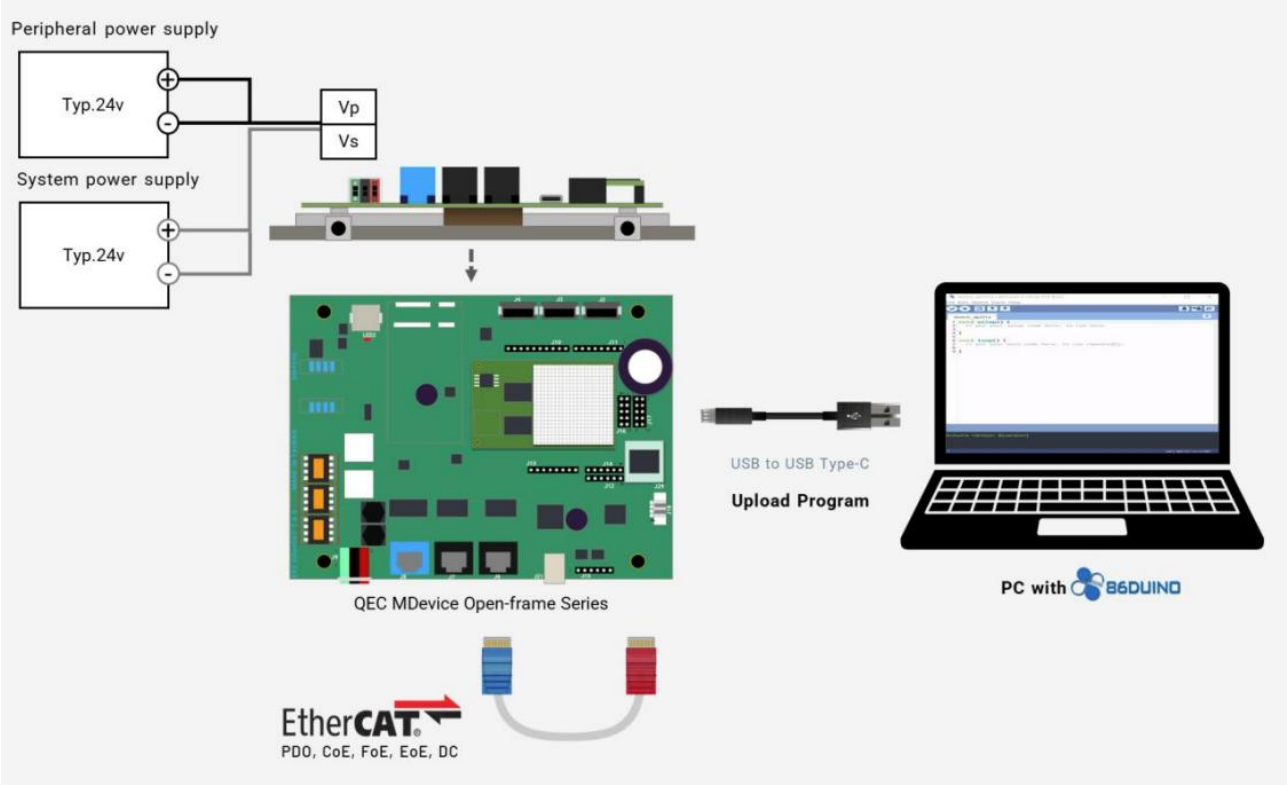
The QEC-M-070T plays multiple roles. Upstream, it connects to the MySQL Server through Ethernet TCP/IP, sends SQL commands, receives results, and writes them back to the HMI. Downstream, it provides a touchscreen HMI for INSERT, READ, DELETE, AUTO Test, LOG, and Settings operations.
This architecture uses SCoop to separate UI updates from background SQL tasks, preventing database operations from blocking the screen and allowing the MySQL test workflow to run reliably on the HMI.

*** Note**
This example is intended for teaching and validation purposes and is suitable for closed local networks or laboratory environments. For production equipment or cross-subnet systems, use a TCP/Socket communication architecture with permission control, packet validation, or encryption.

# 2. Setting Up the QEC-M-070T (Uploading the .ino)

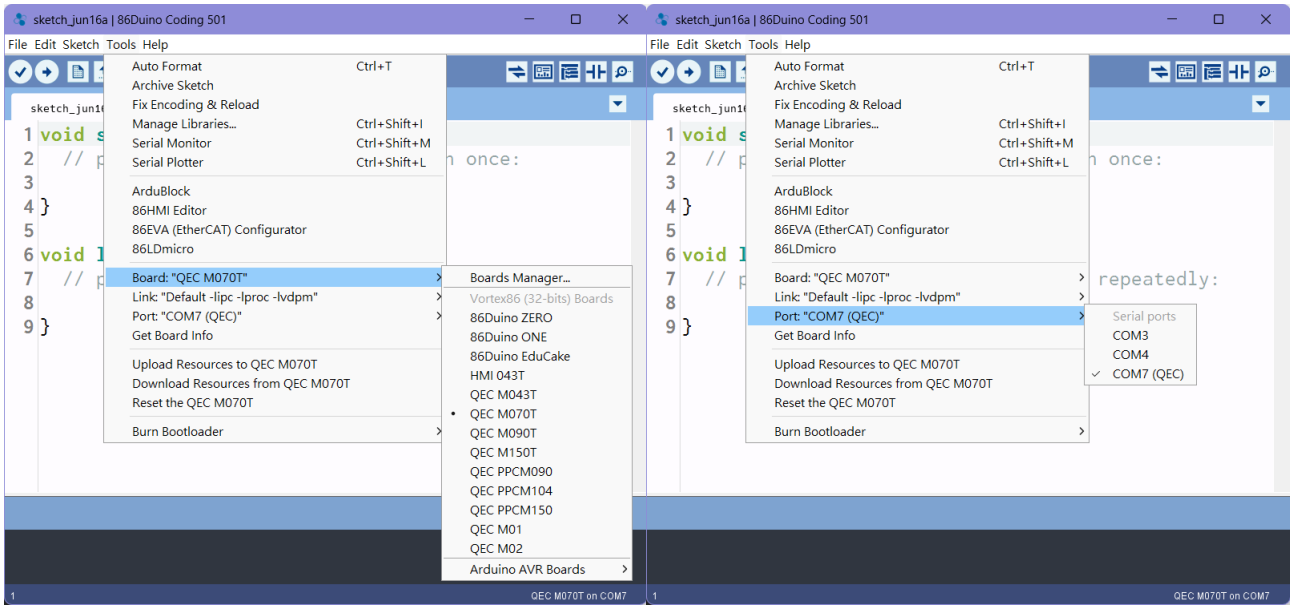
Upload QEC_MySQL_Test_Tool.ino to the QEC-M-070T.

## 2.1 86Duino Environment Setup




Follow the steps below to set up the environment:

1. Connect the QEC-M-070T to your computer using a USB cable. The 86Duino IDE must be installed.
2. Power on the QEC device.
3. Open Device Manager on the computer (press Win+X and select it from the menu), then expand Ports (COM & LPT). You should see “Prolific PL2303GC USB Serial Port (COMx)”. If it does not appear, install the required driver. (The Windows PL2303 driver can be downloaded from the provided link.)
4. Open the 86Duino IDE.
5. Select the correct board: in the IDE menu, choose Tools > Board > QEC-M-070T, or the QEC MDevice model you are using.
6. Select the port: in the IDE menu, choose Tools > Port, then select the USB COM port connected to the QEC MDevice. In this example, it is COM7 (QEC).


sketch_jun16a | 86Duino Coding 501
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
ArduBlock
86HMI Editor
86EVA (EtherCAT) Configurator
86LDmicro
Board: "QEC M070T"
Link: "Default -lipc -lproc -lvdpm"
Port: "COM7 (QEC)"
Get Board Info
Upload Resources to QEC M070T
Download Resources from QEC M070T
Reset the QEC M070T
Burn Bootloader
Boards Manager...
Vortex86 (32-bits) Boards
86Duino ZERO
86Duino ONE
86Duino EduCake
HMI 043T
QEC M043T
QEC M070T
QEC M090T
QEC M150T
QEC PPCM090
QEC PPCM104
QEC PPCM150
QEC M01
QEC M02
Arduino AVR Boards
Serial ports
COM3
COM4
COM7 (QEC)
QEC M070T on COM7

## 2.2 Code Description

The code uploaded to the QEC-M-070T can be divided into five major parts: network settings, HMI, application logic, Task / Runtime tasks, and MySQL communication.

| Layer | Representative Files / Functions | Function |
|---|---|---|
| Network Settings | `IPAddress ip, IPAddress server_addr, mysql_port, g_database_name` | Sets the QEC-M-070T Device IP, MySQL Server IP, Port, and Database. |
| HMI | `build_hmi(), build_page_insert(), build_page_read(), set_*()` | Builds HMI pages, buttons, input fields, and table views, and updates status, logs, and query results. |
| Application Logic | `app_connect(), app_insert(), app_read(), app_delete()` | Receives HMI operation events, validates fields, manages state, handles two-step Delete confirmation, and schedules SQL operations. |
| Task / Runtime | `sqlTask, scoopTask1, loop()` | Separates UI updates from background SQL tasks to prevent MySQL operations from blocking the HMI screen. |
| MySQL Communication | `exec_sql_operation(), exec_insert_raw(), sql_select_into_result()` | Connects to the MySQL Server, executes INSERT / SELECT / DELETE, and organizes SQL results and logs. |

### 2.2.1 Network Settings

At the beginning of the .ino file, confirm that the following settings match your network environment:

```
IPAddress ip(192, 168, 2, 251);       // QEC Device IP
IPAddress server_addr(192, 168, 2, 76); // MySQL Server IP
char user[]     = "qec";
char password[] = "qec";
int mysql_port  = 3307;
char g_database_name[32] = "test";
```

**Note**

The IP address must be on a reachable network segment with the PC / MySQL Server and must not conflict with other devices. Server IP, Port, and Database can be modified in the sketch and re-uploaded, or temporarily adjusted on the HMI SETTINGS page. After reboot, the values return to the sketch defaults.

### 2.2.2 HMI

The HMI defines the user-facing operation interface, including the INSERT, READ, DELETE, AUTO, LOG, and SETTINGS pages. Page creation and switching are handled in hmi_ui.cpp, while the main sketch updates status, query results, and logs through set_*() functions.

```
build_hmi();
build_page_insert(scr);
build_page_read(scr);
build_page_delete(scr);
build_page_auto(scr);
build_page_log(scr);
build_page_settings(scr);
set_insert_status("Ready", COLOR_GREEN);
set_read_status("Ready", COLOR_CYAN);
set_mysql_status_all("Not Connected", COLOR_YELLOW);
```

**Note**
The HMI is responsible only for display and receiving operation events. It does not execute MySQL commands directly.
All MySQL operations are handled by app_*() functions in the main program, keeping screen code separate from database logic.

### 2.2.3 Application Logic

The application logic handles user operations from the HMI, such as CONNECT, INSERT, READ, DELETE, and SETTINGS APPLY. This layer validates input fields, prepares SQL parameters, manages state, and queues SQL operations for background execution.

```
void app_connect();
void app_insert();
void app_read();
void app_delete();
void app_apply_settings();
```

For example, when the user presses INSERT, the program first reads the HMI field values, validates Name, Age, and Class, and then sets the SQL operation type:

```
g_sql_op = SQL_INSERT_HMI;
g_sql_pending = true;
```

**Note**
DELETE uses a two-step confirmation process. A record must first be found and shown in the preview. After pressing DELETE, the user must confirm again before the delete command is actually sent.

### 2.2.4 Task / Runtime

Task / Runtime separates HMI screen updates from MySQL SQL execution. The main loop() maintains screen updates, resets the watchdog, and collects SQL results. sqlTask executes potentially blocking SQL operations, while scoopTask1 handles periodic writes for Auto Test.

```
void loop();

defineTaskLoop(sqlTask) {
  if (g_sql_pending) {
    exec_sql_operation();
    g_sql_pending = false;
    g_sql_done_main = true;
  }
}

defineTaskLoop(scoopTask1) {
  auto_insert_test_row();
}
```

**Note**
MySQL queries or writes may need to wait for network responses, so they must not be executed directly inside HMI callbacks.
This program uses SCoop to place SQL operations in background tasks, allowing the HMI screen to remain responsive during database operations.

### 2.2.5 MySQL Communication

MySQL communication is responsible for connecting to the MySQL Server, sending SQL commands, retrieving query results, and preparing the results for display on the HMI. Major operations include CONNECT, INSERT, READ, FIND, DELETE, VERIFY, RETRY, and CLEAR TABLE.

```
exec_sql_operation();
exec_insert_raw(table, name, age, cls, result);
sql_select_into_result(table, sort, limit, offset, result);
sql_count_rows(table, result);
```

For example, INSERT builds and sends the following SQL command:

```
INSERT INTO test.aaa_test (name, age, class_name)
VALUES ('SAMPLE_001', 60, 'ClassE');
```

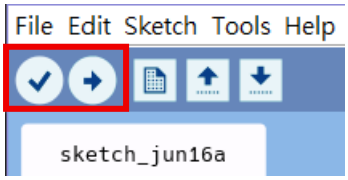
READ queries records according to Table, Sort, Limit, and Page:

```
SELECT id, name, age, class_name
FROM test.aaa_test
ORDER BY id DESC
LIMIT 10 OFFSET 0;
```

**Note**
MySQL communication is responsible only for SQL execution and result organization. It does not update the LVGL screen directly.
After SQL execution completes, the result is returned to the main loop, and the HMI presentation layer updates status, tables, logs, and error messages.

## 2.3 Uploading the Code

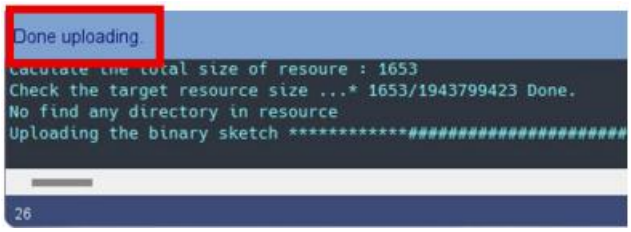
After the code is complete, click the compile button on the toolbar. Once compilation finishes without errors, click Upload.




(The sketch name shown in the figure is for reference only.)

**Note**
Do not close the IDE window or unplug the USB cable during upload. A typical sketch upload takes about 10 to 30 seconds.

A typical sketch upload takes about 10 to 30 seconds. When “Done uploading” appears at the end of the window, the upload is complete. If red error text appears, adjust the code according to the error message. For any other abnormal behavior, please contact us.



# 3. MySQL Server and Table Setup

This tool connects to the test database by default and provides two test table options: aaa_test and bbb_test. Both tables use the same field structure, making it easy to switch between tables during training or testing.

## 3.1 Recommended Table Format

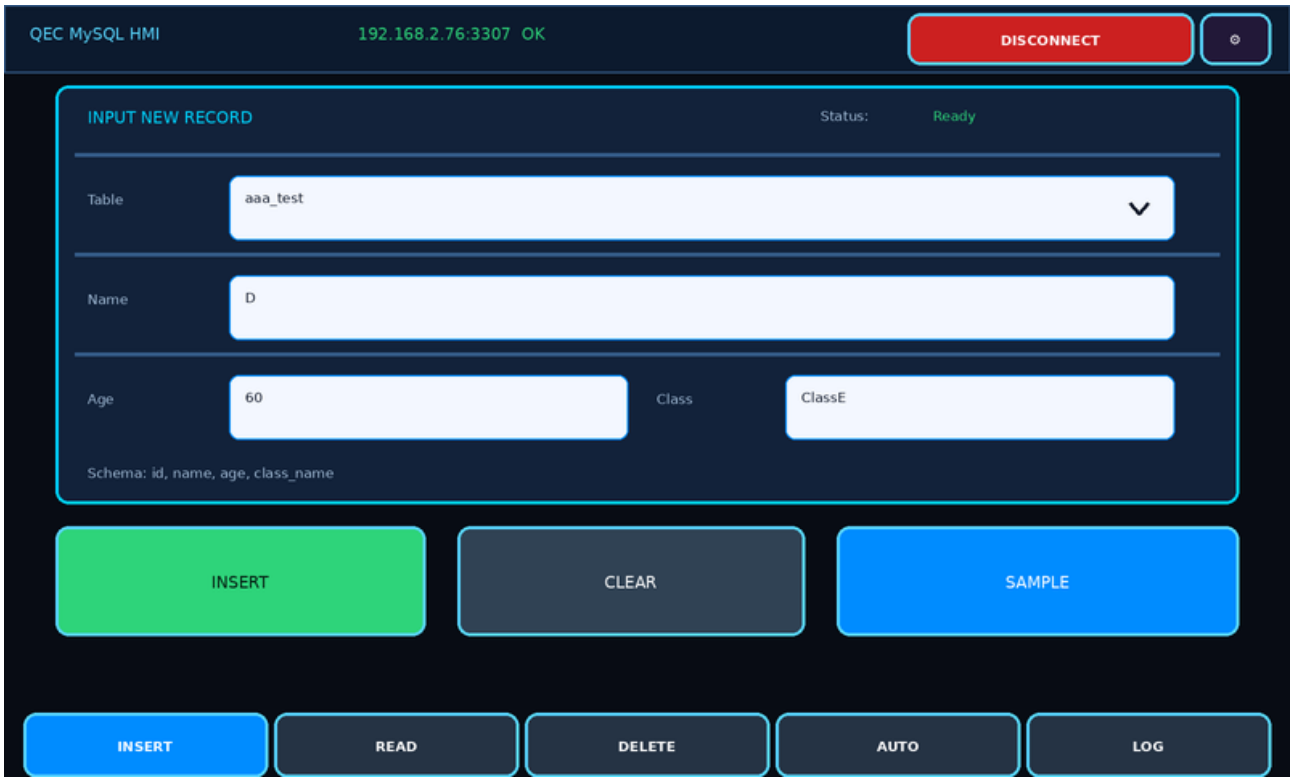
```
CREATE DATABASE IF NOT EXISTS test;
USE test;

CREATE TABLE IF NOT EXISTS aaa_test (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(32) NOT NULL,
  age INT NOT NULL,
  class_name VARCHAR(32) NOT NULL
);

CREATE TABLE IF NOT EXISTS bbb_test LIKE aaa_test;
```

## 3.2 Connection Verification

- The QEC Device IP and MySQL Server IP must be on a reachable network segment.
- The MySQL Server must allow the qec user to connect from the network segment where the QEC device is located.
- If a non-default port is used, enter the new port on the SETTINGS page and press APPLY.
- After CONNECT succeeds, the HMI header changes to the connected state, and the LOG page records the connection event.

# 4. HMI Page Function Description

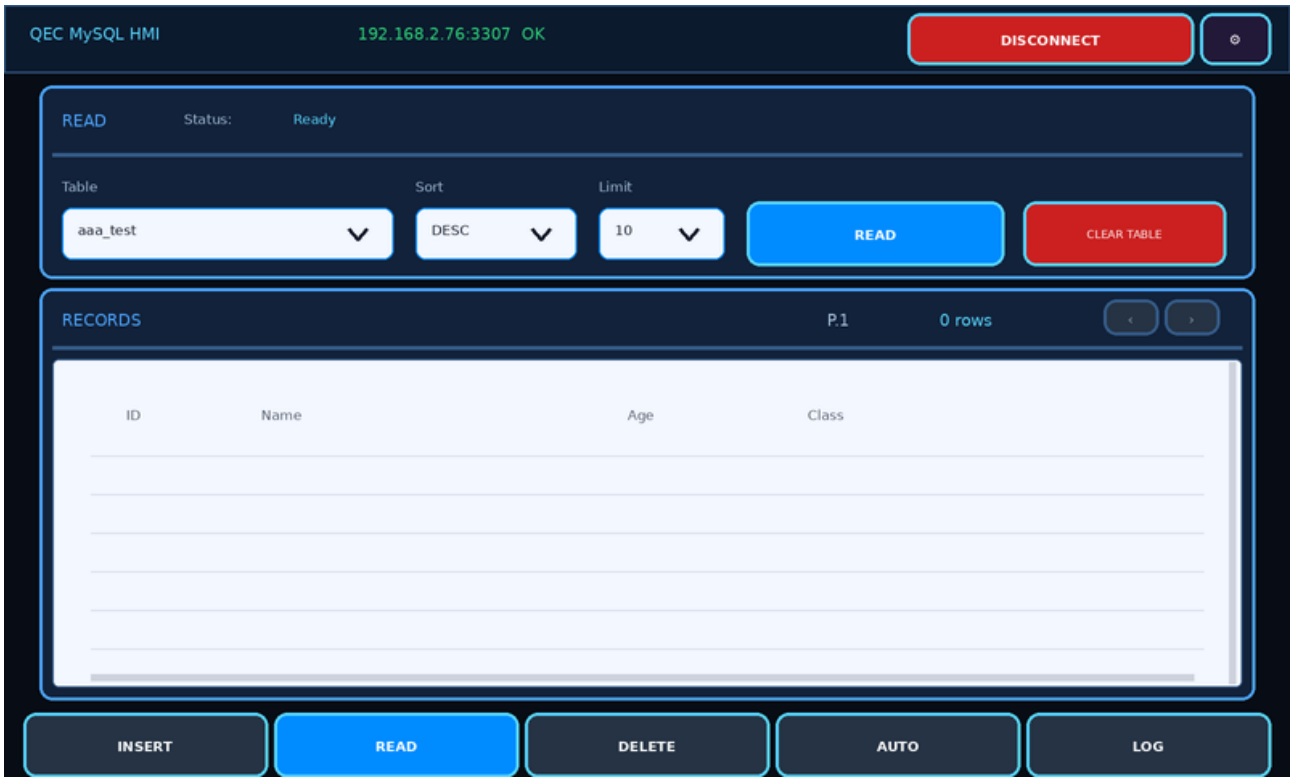
## 4.1 INSERT: Manually Add Test Records




The INSERT page is used to manually create a test record. The user selects a table, enters name, age, and class, and then presses INSERT. The program sends an INSERT INTO command.

| Field / Button | Function |
|---|---|
| Table | Select aaa_test or bbb_test. |
| Name / Age / Class | Enter test data corresponding to the table fields name, age, and class_name. |
| INSERT | Writes one record to MySQL. |
| CLEAR | Clears the input fields. |
| SAMPLE | Automatically generates SAMPLE_### test data for quick testing. |

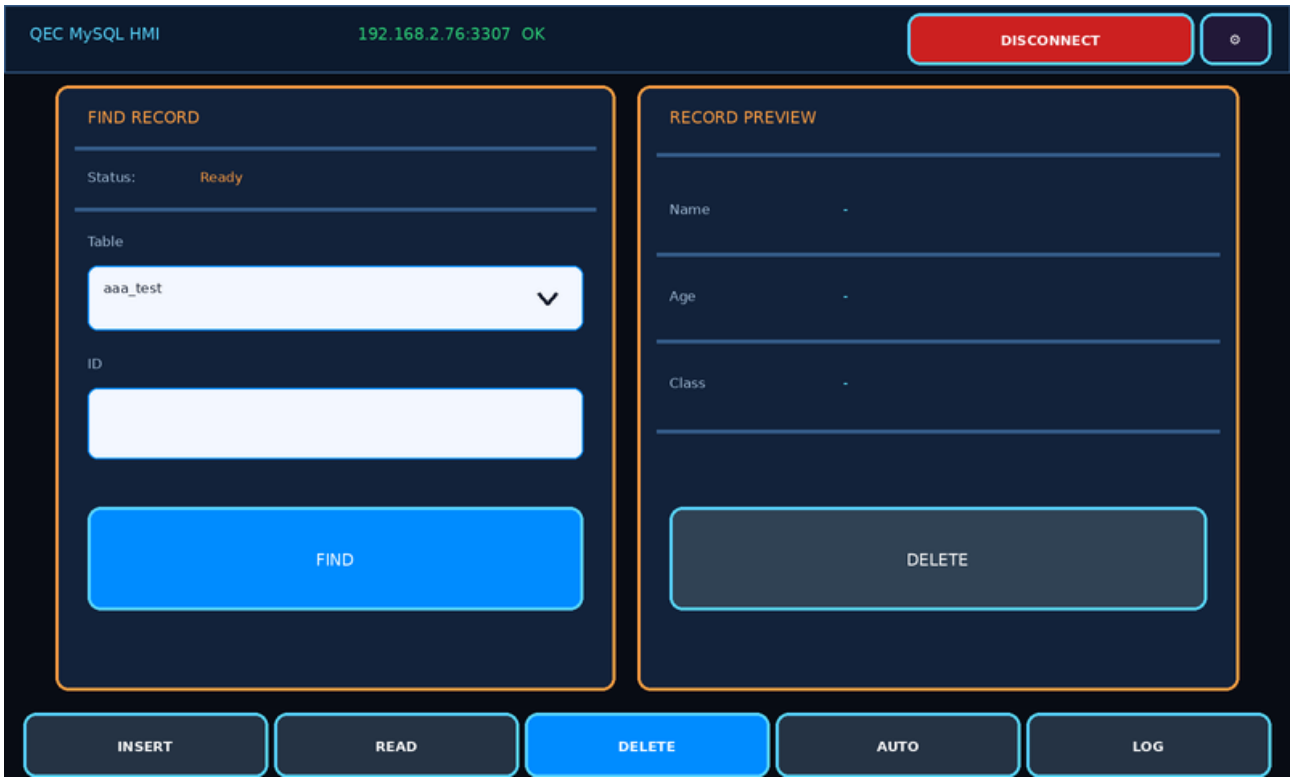
## 4.2 READ: Query Records and Browse Pages




The READ page is used to query table contents. Users can select the sort direction and number of rows per page, and use PREV / NEXT for pagination. The screen shows the current range and total row count, such as 1-10 / 35.

- Sort: ASC / DESC. The default is DESC, making it easier to view the latest records first.
- Limit: 10 or 20. Each query displays up to SQL_MAX_ROWS records.
- CLEAR TABLE: Uses a two-step confirmation design to clear the currently selected table.

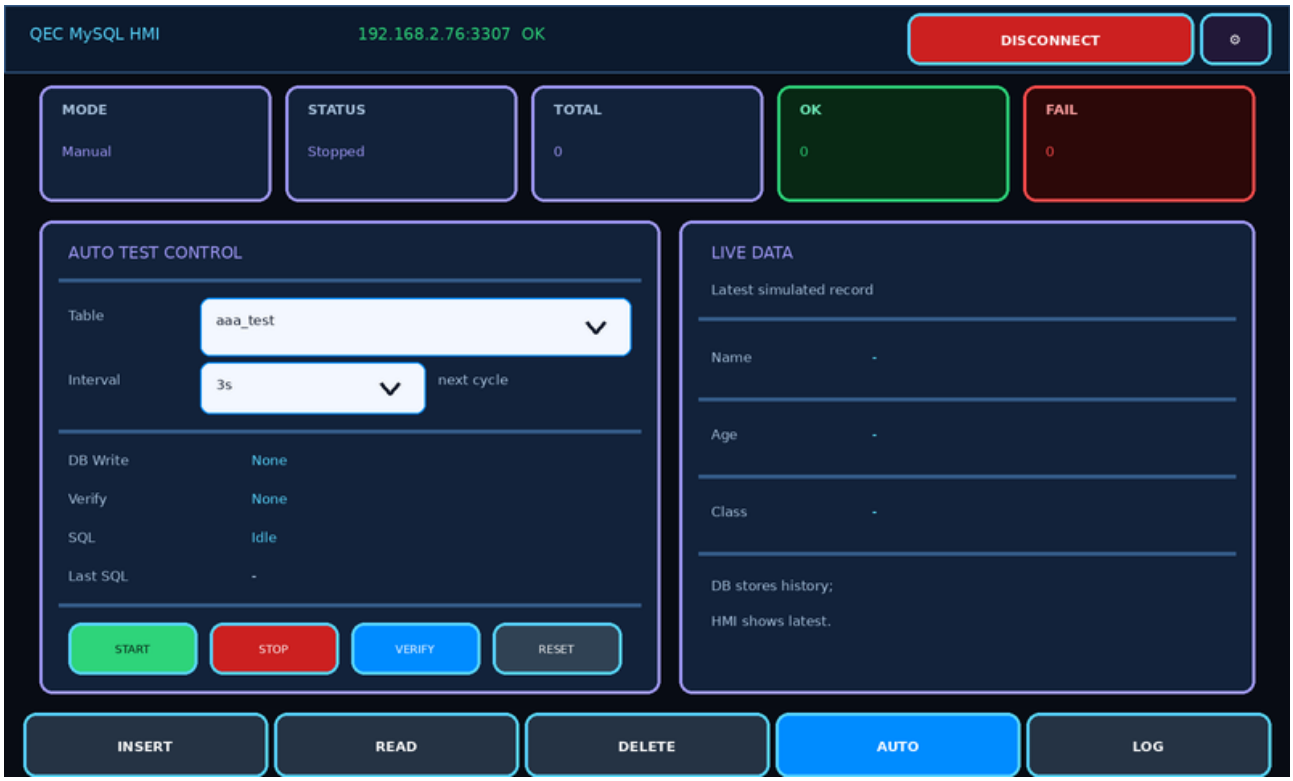
## 4.3 DELETE: Safe Deletion After Lookup




The DELETE page first looks up a record by ID and displays name, age, and class as a preview before deletion is allowed. After the DELETE button enters confirmation mode, it must be pressed again within the countdown period to actually delete the record.

**Safety Design**
DELETE does not delete immediately. It uses the FIND -> PREVIEW -> DELETE -> CONFIRM workflow to reduce the risk of accidentally deleting data during teaching or testing.

## 4.4 AUTO: Automatic Write Test



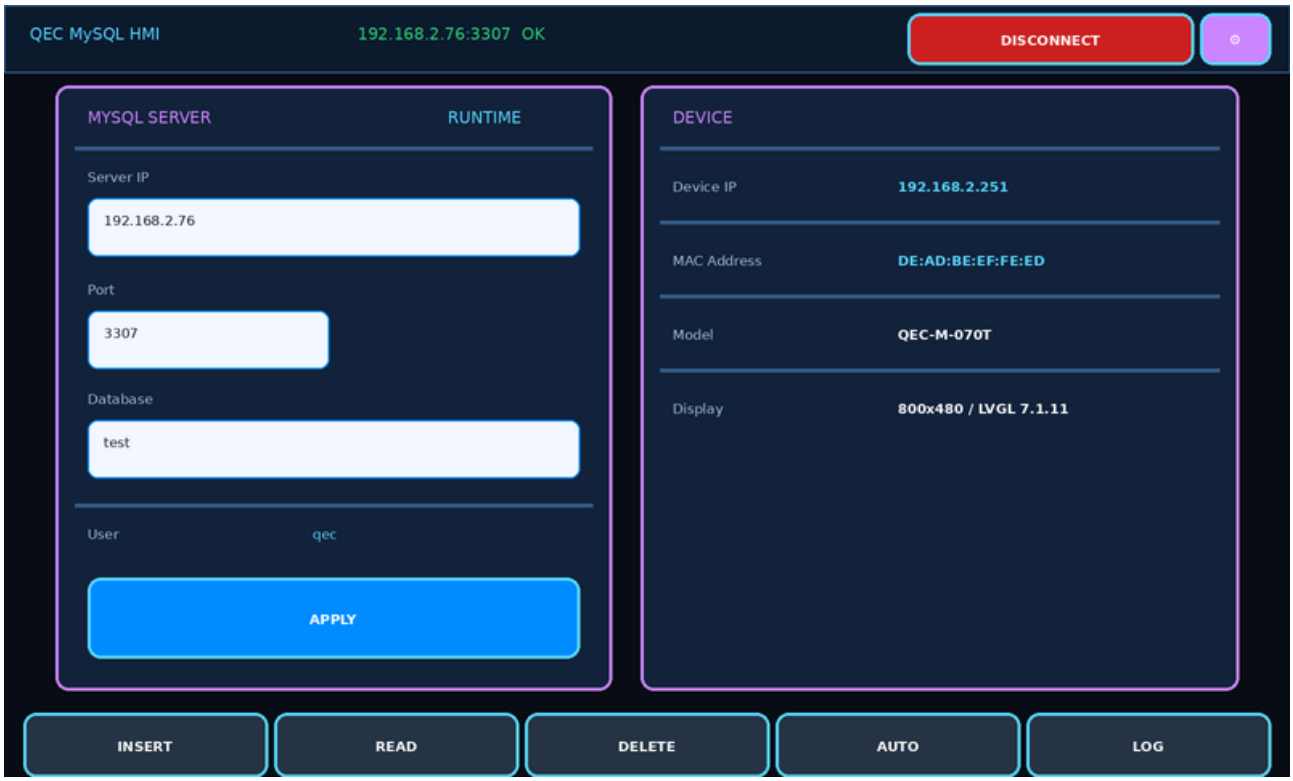

The AUTO page periodically writes simulated data. It is used to observe connection stability, SQL execution time, success/failure statistics, and the latest written record.

- Interval: selectable as 3s, 5s, or 10s.
- TOTAL / OK / FAIL: Displays accumulated Auto Test results.
- VERIFY: Reads back the latest record to confirm that the database contents match the HMI display.
- RESET: Resets the Auto Test counters.

## 4.5 LOG: Operation Log and Retry Buffer


The left side of the LOG page displays HMI operation logs, while the right side shows the failed insert retry buffer. When an automatic write fails, the data is temporarily stored in the buffer, and the user can press RETRY ALL to send it again.

## 4.6 SETTINGS: Runtime Connection Settings




The SETTINGS page allows users to modify the MySQL Server IP, Port, and Database. Pressing APPLY updates the runtime settings and disconnects the current session. The next CONNECT uses the new settings.

# 5. Tutorial: First End-to-End MySQL Test

This chapter combines the previous settings and uses the Purpose / Prerequisites / Steps / Expected Results / Verification / Common Issues structure to guide users through a complete MySQL HMI test.

## 5.1 Purpose

Complete the basic CONNECT -> INSERT -> READ -> DELETE -> AUTO -> LOG workflow, and verify that the QEC HMI and MySQL Server can connect, write, read, delete, and log test status correctly.

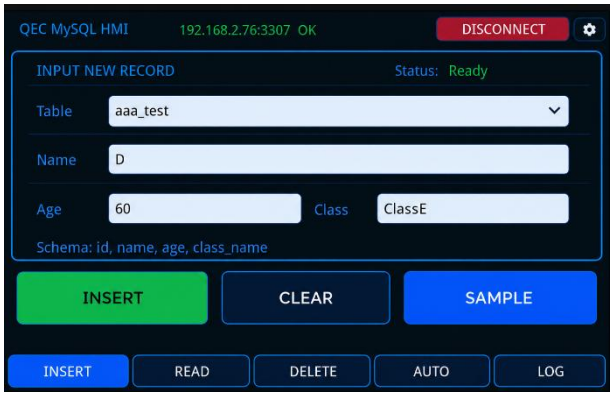
## 5.2 Prerequisites

- QEC_MySQL_Test_Tool.ino has been successfully uploaded to the QEC HMI.
- The QEC Device IP and MySQL Server IP can communicate with each other.
- The MySQL Server has created the test database and the aaa_test / bbb_test test tables.
- The MySQL user qec has SELECT, INSERT, and DELETE permissions.

## 5.3 Steps

1. After booting, enter the INSERT page and confirm that the header status is Not Connected.


2. Press CONNECT and confirm that the status changes to Connected. The LOG page should show a successful connection record.

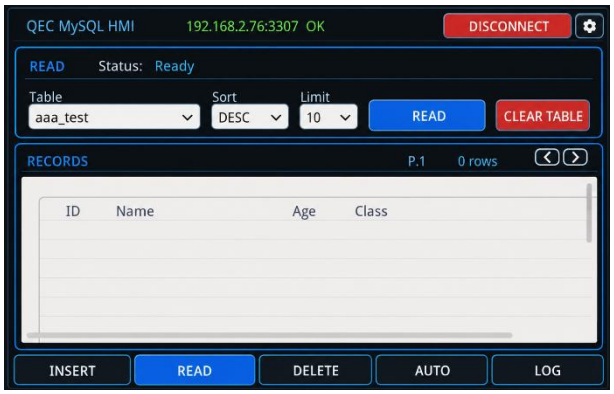




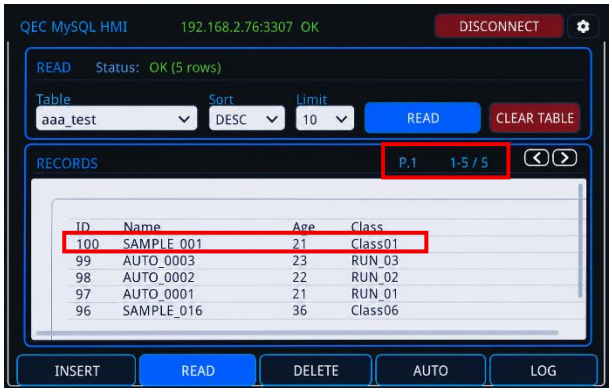
3. On the INSERT page, select aaa_test, press SAMPLE to generate test data, and then press INSERT.



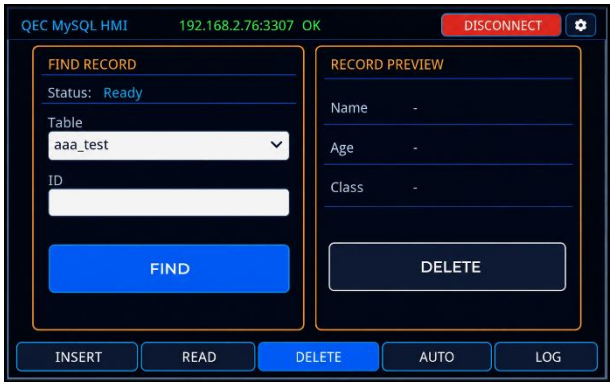
4. Switch to the READ page, select aaa_test, set Sort to DESC, set Limit to 10, and press READ.

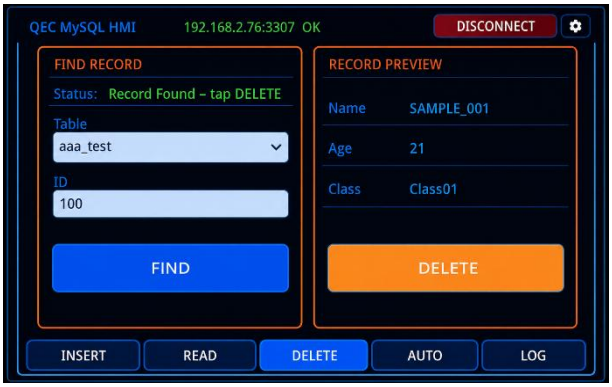



5. Confirm that the newly written record appears in the RECORDS table, and that the row summary displays the current range and total row count.

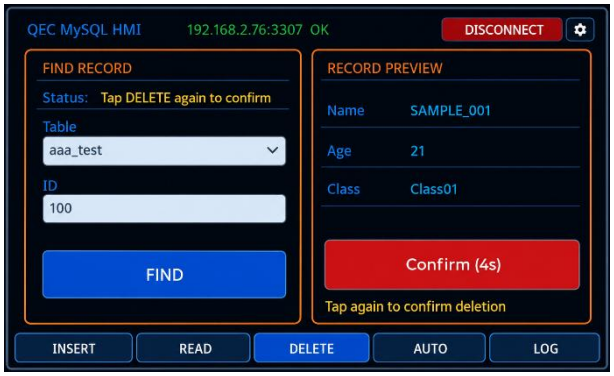


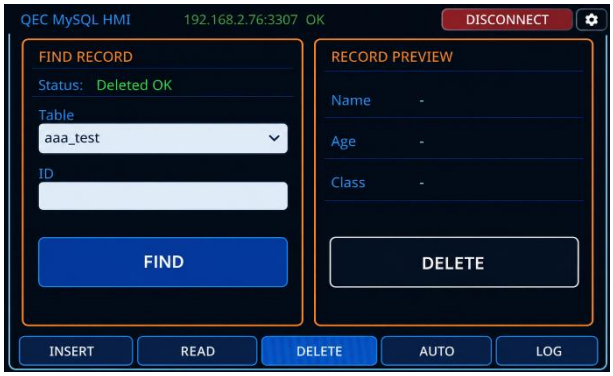

6. Switch to the DELETE page, enter the record ID, press FIND, and confirm that the preview shows the correct data.



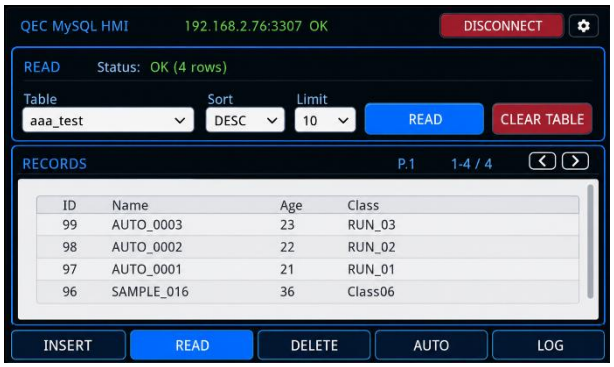



7. Press DELETE and confirm again to delete the record.

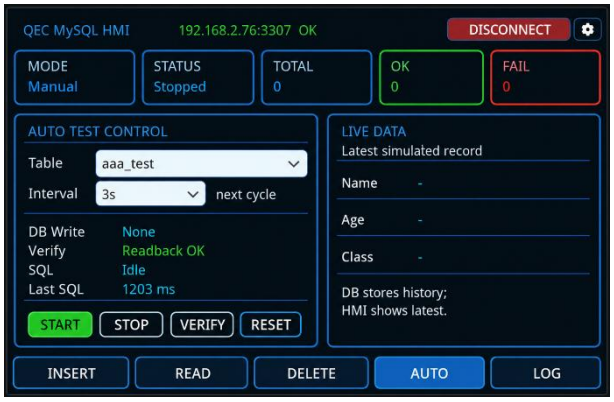


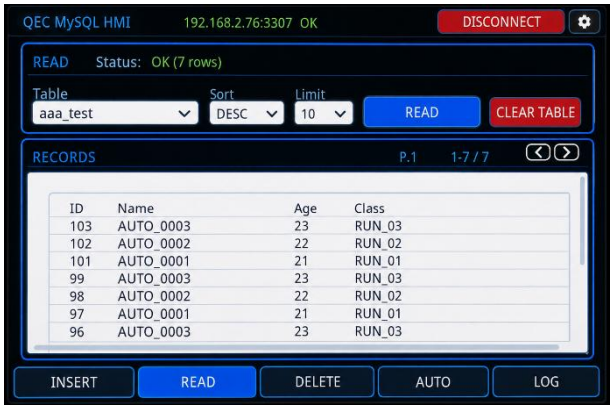

8. Return to the READ page and run READ again to confirm that the record no longer appears in the list.




9. Switch to the AUTO page, select aaa_test and a 3s interval, and press START.




10. Observe TOTAL / OK / FAIL, Last SQL, and the LOG page records. After confirming that automatic writing works normally, press STOP.





## 5.4 Expected Results

- After CONNECT succeeds, the shared connection status across all pages is updated.
- After INSERT succeeds, the new record can be found on the READ page.
- DELETE removes data only after the second confirmation.
- During Auto Test, the OK count should continue increasing, while FAIL remains 0 or can be traced through the Retry Buffer.

## 5.5 Verification

- The SELECT COUNT(*) result in MySQL matches the total count shown on the READ page.
- The LOG page shows records for CONNECT, INSERT, READ, DELETE, AUTO START/STOP, and related operations.
- When disconnection or SQL failure occurs, the HMI status should display an error message without freezing the screen.

## 5.6 Common Issues

- CONNECT failed: Check the Server IP, Port, MySQL user permissions, and firewall settings.
- INSERT failed: Confirm that the table fields are id, name, age, and class_name.
- READ shows no data: Check the selected Table, Database name, and table contents.
- AUTO Test has FAIL records: Switch to the LOG page to check the error, and use RETRY ALL if needed.

# 6. Log and Retry Buffer

HMI LOG uses a fixed line buffer to prevent unlimited text growth during long tests. The Retry Buffer keeps failed automatic write records so users can retry them manually after the connection is restored.

| Item | Purpose | Operation |
|---|---|---|
| HMI LOG | Records connection events, SQL operations, Auto Test status, and error messages. | Press CLEAR LOG to clear on-screen records. |
| Retry Buffer | Stores data from failed automatic writes. | Press RETRY ALL to write again; CLEAR empties the buffer. |
| Last SQL | Displays the elapsed time of the most recent SQL operation. | Used to roughly observe MySQL response speed. |

# 7. Troubleshooting

| Issue | Possible Cause | Solution |
|---|---|---|
| Unable to CONNECT | Incorrect IP/Port, MySQL not running, or firewall blocking the connection. | Check SETTINGS, ping the server, and verify the MySQL port and permissions. |
| INSERT failed | The table does not exist or the field names do not match. | Confirm that aaa_test / bbb_test include id, name, age, and class_name. |
| READ shows 0 rows | The table is empty, or the wrong database/table is selected. | Insert one sample record first, then return to the READ page and query again. |
| DELETE cannot be pressed | No valid ID has been found yet. | Enter an ID and press FIND first. DELETE is enabled only after preview data appears. |
| AUTO FAIL increases | The connection is interrupted or the database is temporarily unavailable. | Check the LOG, restore the connection, and use RETRY ALL. |
| The screen remains usable but SQL is slow | SQL is blocked in sqlTask, while the main loop continues updating LVGL. | Observe Last SQL and reduce the Auto Test frequency if needed. |

# 8. Complete Code and Downloads

The main files corresponding to this guide are listed below.

- QEC_MySQL_Test_Tool.ino: Main program and MySQL logic.
- hmi_ui.cpp / hmi_ui.h: HMI pages and UI API.
- myhmi.cpp / myhmi.h / myeva.cpp / myeva.h: Files generated by the 86HMI Editor.

## 9. Next Steps

This example demonstrates the most basic MySQL connection and functions. Possible extensions for the QEC platform include:

- Add schema check so the HMI can actively verify whether the table fields match the example requirements.
- Provide supplemental documentation for MySQL server installation and permission setup.
- Add more real device data formats, such as temperature, status code, counter, or production line event data.
- For production use, add permission management, SQL parameters, connection keep-alive, and more complete error classification.

## 10. Contact Us

If you encounter any technical issues while following this guide, or if you have purchasing or customization requirements for QEC products, please contact us through any of the following channels:

| | |
|---|---|
| Sales Inquiry / Purchasing | info@qec.tw (QEC product line, such as EtherCAT MDevice) |
| Technical Support / R&D | info@icop.com.tw (ICOP Technology Inc.) |