Using Flowcode to Populate a MySql Database via PHP Script

Introduction

This document is written as a follow up to this post https://www.flowcode.co.uk/forums/viewtopic.php?f=4&t=2201 to show how easy it is by using Flowcode to update your own database(s) with values.

In addition, many may be uncomfortable using a third-party "cloud" hosting services for various reasons so having the ability to host your own server and database may be advantageous. Depending on how you configure things, your server can gather data from devices on your own private LAN or from anywhere in the world via the internet. It should also be possible to have both Host and Client on the same device, such as a Raspberry Pi. The Flowcode in this example has been written and tested for an ESP32-WROOM on the same LAN as the server.

First things first, this is not a tutorial on using MySql, PHP, HTTP or HTML, rather a guide, and is to demonstrate the ease Flowcode makes updating or interacting with the database. This is a forum for Flowcode, not for LAMP/WAMP installs so please post questions relating to such in a more suitable place.

If you don't understand the MySql / PHP commands used here or are a complete novice with LAMP / WAMP installs there are many books and tutorials available. For the complete beginner I can recommend "PHP & MySql in Easy Steps" by Mike McGrath. Although the book is old, it is still a good reference and I used it as guide when I created my first database that Flowcode could update. Also, there are many free online tutorials available such as from "w3schools".

Incidentally my installation is still the same as that book documented and although the versions I'm using are not the most up to date they work fine. I'm running PHP v5.5.9, MySql v5.6.16 and Abyss X1 Server on a W10 machine and all MySql / PHPcode that follows is written for these versions.

If you are a complete novice then I recommend you use the versions above as they have been tested to work as documented. They can be found here

https://aprelium.com/abyssws/download.php
https://www.php.net/releases/
https://downloads.mysql.com/archives/community/

You will need a configured LAMP / WAMP installation, a plain-text editor such as Notepad++ which is a free download, and it is assumed you are familiar with Command Line Interfaces. If you are using newer versions of PHP/MySql then the code may or may not work as typed, but an online resource will most likely explain why and give alternative examples. Newer versions may use differing syntax. UPDATE – See page 11 to use latest MySql and PHP.

If you are a Guru then please feel free to provide better code (with explanations) than the examples here, as I don't claim anything other than basic knowledge and I'm sure there are indeed better ways.

Using Flowcode to Populate a MySql Database via PHP Script

Briefly, MySql is a very popular database application used globally. It does not come with any fancy User Interfaces but you could create your own front end to suit your particular need. Your server application is there to accept and process HTTP requests and PHP provides the interaction with MySql.

If you wish to visit a webpage and you type an address into your browser, your browser then sends the following: GET /URL HTTP/version where URL = the resource you wish to connect with.

Flowcode allows you to easily create these "GET" requests.

We will first look at creating a database, then the PHP code to update it, and once that is running we will look at creating a Flowcode chart that will allow your microcontroller based devices to store their readings directly in your database by calling the PHP script hosted on the server.

Creating your Database

So you plan on having sensors or such like deployed and want to collect, collate and store the data from them. First you need to plan your database. Ultimately, this is the most important part of the process as here you need to establish not just its name, what tables and columns it contains, but also security and validation. It isn't the purpose of this guide to advise on such matters and no security/validation is included. You however should strongly consider such.

Let's say you have some sensors that are gathering values for you. These could be anything gathering whatever. You may have multiple instances of the same type of sensor or even multiple types of sensors with each type providing very different sets to the others.

You will need a way to hold each sensor and the value(s) they gather. It would be nice to timestamp each reading and you might also consider assigning a Primary-Key to uniquely identify each record created too.

A Primary Key is a unique value that can be used to identify a Row (record) in your database.

Suppose you are deploying two different types of sensors, then you could assign a table to each with each table having fields appropriate to the values you wish to record.

You might decide on:-

Database = my_db

Table(s) = sensor_a and sensor_b

With the table for sensor_a being

ID	Timestamp	Sensor	Value
1	15-11-2023 16:00:00	1	17
2	15-11-2023 16:05:24	2	27
3	15-11-2023 16:08:14	1	24

Using Flowcode to Populate a MySql Database via PHP Script

In the above table, ID is the Primary Key providing a unique reference for each new record, Timestamp is added when the record is created with Sensor an alphanumeric representing the name of a device and Value being the actual integer data sent. As MySql can automatically insert the information in the first two columns for us, our sensors using Flowcode only need to send the sensor name and value (key-pairs).

Note you can define your own data types for yourself, this is only an example.

Creating the Database

We will create the database, add a User with Privileges then create the above table. Open the MySql Command Line Interface. Useful keys when working in the CLI include

F3 – insert last line entered (paste what you previously typed)

Up /Down – scrolls through the previously typed entries

From the MySql Command Line Interface prompt (mysql>) enter

```
mysql> CREATE DATABASE IF NOT EXISTS my_db ;
```

(The above line creates a database called my_db)

Add User(s) and Privileges (see UPDATE on page 11 if using latest versions)

```
mysql> GRANT SELECT, INSERT, UPDATE ON my_db.* TO ‘user’@’localhost’ IDENTIFIED BY ‘1234’ ;
```

(The above line creates a username on my_db of user with a password of 1234)

To use the new database you created you first need to tell MySql to use it

```
mysql> USE my_db ;
```

That’s it. You now have a new database called my_db

You can check by running the following commands

```
mysql>SHOW DATABASES ;
```

```
mysql>SHOW GRANTS FOR ‘user’@’localhost’ ;
```

Next we will create the table to hold our information.

Using Flowcode to Populate a MySql Database via PHP Script

Creating the Table (see UPDATE on page 11 if using latest versions)

To create the sensor_a table documented above we will issue the following command.

```
mysql> CREATE TABLE IF NOT EXISTS sensor_a ( ID INT AUTO_INCREMENT PRIMARY KEY, Timestamp TIMESTAMP, Sensor VARCHAR(10), Value INT ) ;
```

where ID = name of 1st Column

INT = numeric data type for 1st column

AUTO_INCREMENT = automatically increment 1st column upon creation

PRIMARY KEY = unique reference for each new record

Timestamp = name of 2nd column

TIMESTAMP = automatically inserted date/time in 2nd column based on server time settings

Sensor = name of 3rd column

VARCHAR(10) = Variable length string (10 characters maximum) for 3rd column

Value = name of 4th column

INT = numeric data type for 4th column

You can check the above by issuing the following command

```
mysql>EXPLAIN sensor_a ;
```

That’s it. You have now created a database and table to hold your readings. You could create a table for sensor_b in a similar fashion.

Next we need to create scripts to link everything together.

Using Flowcode to Populate a MySql Database via PHP Script

Creating a PHP script to interact with my_db

Before PHP can interact it needs to connect with your database and this is taken care of by a script. This script will connect using the user/password created above. However as this script contains login information it should not be stored in the same folder as your HTML documents. If using Abyss as your web server the default folder for your HTML is C:\Abyss Web Server\htdocs so perhaps use C:\Abyss Web Server\

Using a plain text editor such as Notepad++ create a new document and enter the following (note it is recommended you type all the following code examples directly as pasting can introduce issues):-

```
<?php
$dbc = @mysqli_connect
( ‘localhost’ , ‘user’ , ‘1234’ , ‘my_db’ )
OR die
( mysqli_connect_error() ) ;
mysqli_set_charset( $dbc , ‘utf8’ ) ;
?>
```

Save the above in the parent directory as connect.php

If using Abyss this will be C:\Abyss Web Server\

We will call this script from within another, whenever we wish to interact with the my_db database.

Using Flowcode to Populate a MySql Database via PHP Script

Creating a PHP script to Update Database my_db Table sensor_a

Using Notepad++ or similar create a new document and enter the following:-

```
<?php

# Script to update table sensor_a using GET.

# Only allow GET method.

if ( $_SERVER[ 'REQUEST_METHOD' ] == 'GET' )

{

# Connect to MySql Database my_db.

require( '../connect.php' ) ;

# Create temporary variables from SuperGlobals

$Sensor = mysqli_real_escape_string( $dbc, $_GET['Sensor']);

$Value = mysqli_real_escape_string( $dbc, $_GET['Value']);

# Create and execute MySQL query to insert new record into database "my_db", table "sensor_a".

# Query =

$q = "INSERT INTO sensor_a

( Sensor , Value )

VALUES ( ‘$Sensor’ , $Value ) " ;

# Execute Query

mysqli_query( $dbc , $q ) ;

# Close connection.

mysqli_close( $dbc ) ;

}

# Job Done = Enjoy a well earned beer <s>

?>
```

Save the above in C:\Abyss Web Server\htdocs folder as update_sensor_a.php

Using Flowcode to Populate a MySql Database via PHP Script

Testing the above Script

Useful CLI commands for MySql include

SHOW TABLES ;	this shows a list of tables in the current database
EXPLAIN <table-name> ;	this provides details of all columns in the table
SELECT * FROM <table-name> ;	this shows all records in the table

To test what we have done above open a browser and type the following into the address bar

Localhost/update_sensor_a.php?Sensor=1&Value=17 <enter>

(alternatively: 127.0.0.1/update_sensor_a.php?Sensor=1&Value=17 <enter>)

Where	localhost/	= host address to connect with
	update_sensor_a.php	= script we wish to run
	Sensor=1	= key-pair to update Sensor column with the value of 1
	Value=17	= key-pair to update Value column with the value of 17

Now at your MySql Command Line Interface enter the following

```
mysql>SELECT * FROM sensor_a ;
```

If all went well you should now have one line in your table containing ID number / Timestamp / Sensor number / Value with Sensor and Value being 1 and 17 respectively.

Have a play using your browser. Send sensor and value key-pairs and also familiarise yourself with CLI commands. However please do not change anything to do with the table structure yet as the next section will deal with updating from a microcontroller.

If the above does NOT work there is no point in going further until it does. Check your typing as that will most likely be the problem. Also if not using Abyss as your server you will need to establish where to store the “connect” and “update” files. If using later software versions then perhaps seek online help to update code.

Please note that we have not addressed security, validation, error handling or even anything in the way of returns. These are all things you should consider and there are many examples of such out there.

Before we move on we need to take note of the server’s IP address. Although we used localhost or 127.0.0.1 as the address in our browser, these are just local to your machine and cannot be used to reach it from elsewhere on your LAN. In Windows you can see the address you are using from Windows Command Prompt by entering ipconfig which will return your network details. Take note as we will need them later (e.g. IP Address = 192.168.1.123). It is worth noting that your server requires a static IP address so that the IP address does not change.

Using Flowcode to Populate a MySql Database via PHP Script

Creating a Flowcode Chart to update the Database

Using Flowcode v10 we will create a chart to update the database we created above (my_db / sensor_a).

We will need a way to connect to the same LAN as your server and Flowcode provides components to enable this from WiFi enabled targets such as the ESP32 to external boards such as the ESP8266.

The example shown here is written using Flowcode v10 for the free ESP32-WROOM target. With Flowcode though, it is very easy to choose a different chip as it is target independent.

Rather than use an actual sensor in this example, we will instead just include a User Macro that generates a random integer value. You could insert a routine to obtain whatever data you wish though.

Briefly, we will

- Initialise components
- Connect to WiFi
- Enter a loop
- Obtain sensor data
- Convert data into String values
- Create a String to send that will include the script to use and key-pairs
- Obtain the length of above String
- Connect to the server
- Tell server length of String we are sending
- Send String
- Close the connection
- Wait before repeating

In the attached Flowcode chart you will need to first provide the following information for the variables listed below:

Variable	**Value**
SSID	Enter your WiFi SSID
Pword	Enter your password for above
Server	Enter the IP address of your server (e.g. 192.168.1.123). Note you can also use URLs
Script	Name of script to be run (e.g. update_sensor_a.php)
Sensor	Unique name for your sensor/device. Each instance of Flowcode deployed should have a unique reference to differentiate between them. This can be alphanumeric.

Note: Script and Sensor have preconfigured values in the example but you should modify to suit.

Using Flowcode to Populate a MySql Database via PHP Script

When we tested our database and scripts above, we used a browser to send:-

Localhost/update_sensor_a.php?Sensor=1&Value=17

Now we will use Flowcode to create and send the following string:-

"GET /update_sensor_a.php?Sensor=*Sensor*&Value=*Value* HTTP/1.1\r\nHost: *server*\r\n\r\n"

Where

GET	=	is the request method
update_sensor_a.php	=	is the script we will call
Sensor=*Sensor*	=	is the first key-pair value
Value=*Value*	=	is the second key-pair value
HTTP/1.1\r\nHost: *Server*\r\n\r\n	=	is required HTTP

Note that we can send as many key-pairs as required, they are just separated by an “&” but be sure to increase the size of the “Send” string to accommodate.

With reference to the attached chart Update_my_db you can see we only need two components to update the database

WLAN ESP32	Component Libraries>Comms>WLAN (ESP32) (2D)
Network Comms	Component Libraries>Comms>Network Communications (2D)

If you look in Network Comms Properties you will see we linked it to the ESP32 and here you can also set your simulation details if required (if set correctly then you can simulate the chart from within Flowcode without the need of compiling to target).

The first comment in the chart is a reminder to set the required variables. In the example the variable Script has been preconfigured to update_sensor_a.php and the variable Sensor to FC1 you can change these to suit.

We then initialise the components before attempting to join your WiFi network. We will keep looping until we successfully connect as defined by the Ret value of “1” before entering the Main Loop of the program (note if simulating you can enter “1” in the Simulation Debugger for variable “Ret” which will allow your chart to progress).

We then branch to a User Macro (Get_Data) in which we will gather whatever information is required to be sent to the database.

In the example we will just use a calculation box to generate a random number, but this could be anything such as obtaining an ADC reading or sampling some sensor.

Importantly, whatever you gather must be converted into a String before we can send, and so we do this conversion before exiting the Macro.

Now back in the Main chart we then build our string to send. Flowcode occasionally has issues when concatenating multiple strings at once so we will create ours step by step and once created we will then calculate the length of the string.

The chart then opens a socket and attempts to connect with the server on port 80. Note that although Flowcode allows you to test for success in these actions we have not incorporated here and you should consider such in any deployment.

We include a short delay to allow time for the server to respond. Depending on latency this could be many hundreds of milliseconds and delays between 200 and 450mS are not unreasonable. If everything is on your own LAN then these can be drastically reduced.

Next we send our string with another short delay before closing our socket.

The chart then has a delay of 30s before repeating but you could perhaps have your chip sleep or such like.

Before compiling to target remember to go to Project Options and set the Programmer Port your ESP is connected to.

Once running (or simulating), again issuing the following command at your MySql CLI should show updated values

```
mysql>SELECT * FROM sensor_a ;
```

The above is only intended as a guide to getting things working and I hope it is of some help. Any actual deployment should consider the points made above regarding security, validation, errors and tests.

Thanks go to RGV250 for his help in the creation and testing of this guide.

Using Flowcode to Populate a MySql Database via PHP Script

UPDATE if using latest MySql and PHP

To use the latest versions of MySql and PHP

1) If updating from previous versions, uninstall all previous versions and run a registry cleaner
2) Do a clean install

There are a few differences between versions, but we are only concerned with these steps.

After we create the database my_db on page 3 we have to use it

```
mysql>USE my_db ;
```

We now create a new user by entering the following

```
mysql>CREATE USER ‘user’ IDENTIFIED BY ‘1234’ ;
```

(The above line creates a username of user with a password of 1234)

On page 4 when creating the table sensor_a use the following command

```
mysql> CREATE TABLE IF NOT EXISTS sensor_a ( ID INT AUTO_INCREMENT PRIMARY KEY, Timestamp DATETIME DEFAULT CURRENT_TIMESTAMP, Sensor VARCHAR(10), Value INT ) ;
```

We can then grant privileges by issuing this command

```
mysql>GRANT SELECT, INSERT, UPDATE ON sensor_a  TO ‘user’ ;
```

(The above line assigns privileges to user on table sensor_a)

The rest of the guide should work as documented.

Using Flowcode to Populate a MySql Database via PHP Script

Retrieving results from the database

Now we can populate the database it would be good if we could retrieve values from it too. Again note that we have not addressed security, validation or error handling. These are all things you should consider and there are many examples of such out there

In the previous examples we have a sensor called FC1 that updates a table called sensor_a within the my_db database. If you have been playing, your database should contain many differently named sensors and associated values. If not, use your browser to add in a few more values for FC1 and also add some differently named sensors.

Once you have added more data, make sure the last few updates are not named FC1. If you then display the table contents (mysql>SELECT * FROM Sensor_a;) you will see a list of entries with sensor FC1 appearing up from the bottom.

Let's decide that we want the last value entered by a sensor, as we may want to act on the most recent value. Remember that every addition to the database is automatically timestamped so that makes things easier for us.

To get the latest value entered for our sensor FC1 we need to search the table for all entries containing FC1, then arrange them by descending time extracting the latest update before sending the value back. Lets say we want to return the sensor name and value such as sensor=xxxx

We can do this with a PHP script that will accept our GET request, interact with the database to extract the information and send it to us.

As before we will send a key-pair in a GET statement but instead of updating the database it will retrieve information

Using Flowcode to Populate a MySql Database via PHP Script

Using Notepad++ create a new document and enter the following

```
<?php
# Script to request latest sensor value table sensor_a using GET.
# Only allow GET method.
if ( $_SERVER[ 'REQUEST_METHOD' ] == 'GET' )
{
# Connect to MySql Database my_db.
require( '../connect.php' ) ;
# Create temporary variables from SuperGlobals
$Sensor = mysqli_real_escape_string( $dbc, $_GET['Sensor']);
# Create and execute MySQL query to retrieve record from database "my_db", table "sensor_a"
# Query =
$q = "SELECT Value FROM sensor_a WHERE Sensor = '$Sensor' ORDER BY Timestamp DESC LIMIT 1" ;
# Execute Query
$result = mysqli_query( $dbc , $q ) ;
if (mysqli_num_rows($result) > 0) {
    // OUTPUT DATA OF EACH ROW
    while($row = mysqli_fetch_assoc($result)) {
       echo $Sensor, "=",$row["Value"] ;
    }
  } else {
    echo "0 results";
  }
# Close connection.
mysqli_close( $dbc ) ;
}
?>
```

Save the above in htdocs as request_sensor_a.php

Testing the above Script

To the test the above script open your browser and enter

Localhost/request_sensor_a.php?Sensor=FC1

This should return

FC1=xxxx where xxxx is the value.

Again using your browser now enter

Localhost/update_sensor_a.php?Sensor=FC1&Value=12345

This creates a new entry for sensor FC1 with a value of 12345

If we again run

Localhost/request_sensor_a.php?Sensor=FC1

This should return

FC1=12345

If it doesn't then double check your new script for spelling mistakes and errors. Once you have it working have a play. Update your database then retrieve values for different sensors.

Using Flowcode to Populate a MySql Database via PHP Script

Creating a Flowcode Chart to retrieve values from the Database

This is just an example to illustrate how you *could* receive and process a request using Flowcode. The example is written for an ESP32-Lolin board which has an onboard LED connected to pin 22. We will poll the database at intervals and if the returned value exceeds a predefined "threshold" value then the LED will illuminate. If the value is below, then the LED will extinguish.

Briefly, as before we will

- Initialise components
- Connect to WiFi
- Enter a loop
- Create a String to send that will include the script to use and a key-pair
- Obtain the length of above String
- Connect to the server
- Tell server length of String we are sending
- Send String

However in addition to the above we will then

- Receive reply from Server
- Close the connection
- Parse the reply obtaining our data
- Do something with the data
- Wait before repeating

Although when we use the browser to make the request, we get the reply FC1=12345 (sensor name and value), in reality far more HTML is sent by the server. In the attached chart we capture the response in a variable called Rx_String and we use the variable Rx_length to obtain the length of the string, which is in excess of 200 bytes.

We will need to parse this string to obtain our data and there are many ways to do this. Again this is just an example and will use a Circular Buffer to "LookForValue(s)" and also a loop to extract values.

First we should check that we connected and received a page by looking for "200 OK" in our received string and if so continue. Note this just indicates we successfully connected and obtained "some" reply from the server.

Next we should confirm that we have our expected Sensor in the reply and if so we can then loop extracting our value. This will still be in string format so we will also convert to an integer value.

In the chart we include a LED that only comes on if the received value exceeds a predefined Threshold value. In the example it will come on if the returned value of FC1 is greater than 17. You can modify to suit.

In the attached Flowcode chart you will need to first provide the following information for the variables listed below:

Using Flowcode to Populate a MySql Database via PHP Script

Variable	Value
SSID	Enter your WiFi SSID
Pword	Enter your password for above
Server	Enter the IP address of your server (e.g. 192.168.1.123). Note you can also use URLs
Script	Name of script to be run (e.g. request_sensor_a.php)
Sensor	Unique name for your sensor/device. Each instance of Flowcode deployed should have a unique reference to differentiate between them. This can be alphanumeric.
Threshold	Trigger level to "do something". This can be any value you wish

Note: Script, Sensor and Threshold have preconfigured values in the example but you should modify to suit.

If simulating then in the Simulation Debugger you can also view the variables as the chart progresses.

Testing the Flowcode Chart

As mentioned previously, we have a Threshold value for FC1 and if this is exceeded the LED will come on.

Using a browser enter the following

Localhost/update_sensor_a.php?Sensor=FC1&Value=10

This creates a new entry for sensor FC1 with a value of 10.

Localhost/request_sensor_a.php?Sensor=FC1

This should return FC1=10

Connect / run your Flowcode chart and wait at least 30 seconds (or however long your loop delay is). All being well nothing visible should happen, the LED should remain OFF

Now using a browser enter the following

Localhost/update_sensor_a.php?Sensor=FC1&Value=27

This creates a new entry for sensor FC1 with a value of 27. The next time the Flowchart retrieves a value for FC1 the LED should turn ON. It will stay on until the retrieved value for FC1 is less than (or equal to) the Threshold (17).

You can test this by using your browser (or FC powered device) to update the database with values for FC1 above and below the Threshold.