**EMBEDDED SYSTEM OPERATING SYSTEM DESIGN AND IMPLEMENTATION**

Homework2

Temp.Data

Student Name: Le Minh Hai Student ID: M0732772

Student Name: Nguyen Thanh Tuan Student ID: P0731604

1. **Operation Description**

This project will collect data from the real environment (temperature) and store temperature value into database using MySQL. Finally, my group will show this in to web sever as Google line chart by using PHP JSON data

1. **Source Code with Comment**

In this project, we have 3 main source codes:

1. Building Databases

Begin with Terminal in Raspberry Pi:

$mysql -u root -p

After type password, we can using mysql:

mysql>show database;

>create database testing;

>use testing

>create table tbl\_sensors\_data (`sensors\_data\_id` int(11) NOT NULL, `sensors\_temperature\_data` varchar(30) NOT NULL, `sensors\_data\_date` date NOT NULL, `sensors\_data\_time` time NOT NULL) ENGINE=InnoDB AUTO\_INCREMENT=484 DEFAULT CHARSET=latin1;

**##Create fake database for testing**

> insert into tbl\_sensors\_data`(`sensors\_data\_id`, `sensors\_temperature\_data`, `sensors\_data\_date`, `sensors\_data\_time`) VALUES

(339, '28', '2017-08-08', '10:00:00'),

(340, '26', '2017-08-07', '10:10:00'),

(341, '36', '2017-08-06', '10:20:00'),

……………………………………,

(482, '34', '2017-03-24', '09:50:00'),

(483, '30', '2017-03-23', '10:00:00');

**##Auto increase 1 after this data have input, begin with 484**

> ALTER TABLE tbl\_sensors\_data ADD PRIMARY KEY (`sensors\_data\_id`)

>ALTER TABLE tbl\_sensors\_data MODIFY `sensors\_data\_id` int(11) NOT NULL AUTO\_INCREMENT,AUTO\_INCREMENT=484;

1. PHP code

This code will show data from database as a line chart.

Index.php (/var/www/html/index.php)

<?php

//index.php

$connect = mysqli\_connect("localhost", "root", "**PASSWORD**", "testing");

$query = '

SELECT sensors\_temperature\_data,

UNIX\_TIMESTAMP(CONCAT\_WS(" ", sensors\_data\_date, sensors\_data\_time)) AS datetime

FROM tbl\_sensors\_data

ORDER BY sensors\_data\_date DESC, sensors\_data\_time DESC

';

$result = mysqli\_query($connect, $query);

$rows = array();

$table = array();

$table['cols'] = array(

array(

'label' => 'Date Time',

'type' => 'datetime'

),

array(

'label' => 'Temperature (°C)',

'type' => 'number'

)

);

while($row = mysqli\_fetch\_array($result))

{

$sub\_array = array();

$datetime = explode(".", $row["datetime"]);

$sub\_array[] = array(

"v" => 'Date(' . $datetime[0] . '000)'

);

$sub\_array[] = array(

"v" => $row["sensors\_temperature\_data"]

);

$rows[] = array(

"c" => $sub\_array

);

}

$table['rows'] = $rows;

$jsonTable = json\_encode($table);

?>

<html>

<head>

<script type="text/javascript" src="https://www.gstatic.com/charts/loader.js"></script>

<script type="text/javascript" src="//ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

<script type="text/javascript">

google.charts.load('current', {'packages':['corechart']});

google.charts.setOnLoadCallback(drawChart);

function drawChart()

{

var data = new google.visualization.DataTable(<?php echo $jsonTable; ?>);

var options = {

title:'Sensors Data',

legend:{position:'bottom'},

chartArea:{width:'95%', height:'65%'}

};

var chart = new google.visualization.LineChart(document.getElementById('line\_chart'));

chart.draw(data, options);

}

</script>

<style>

.page-wrapper

{

width:1000px;

margin:0 auto;

}

</style>

</head>

<body>

<div class="page-wrapper">

<br />

<h2 align="left">Group9: Welcome</h2>

<h2 align="center">Display Temperature</h2>

<div id="line\_chart" style="width: 100%; height: 500px"></div>

</div>

</body>

</html>

1. Python code

We write this code in python3. This code will send date, time and temperature to **testing** database (**tbl\_sensors\_data** table). We update the value every 10 seconds.

**Get\_temp\_sql.py**

import os

import time

import datetime

import glob

import pymysql

pymysql.install\_as\_MySQLdb()

import MySQLdb

from time import strftime

import RPi.GPIO as GPIO

#define GPIO pins as input

GPIO.setmode(GPIO.BOARD)

GPIO.setup(7, GPIO.IN)

GPIO.setup(11, GPIO.IN)

GPIO.setup(12, GPIO.IN)

GPIO.setup(13, GPIO.IN)

GPIO.setup(15, GPIO.IN)

GPIO.setup(16, GPIO.IN)

GPIO.setup(18, GPIO.IN)

GPIO.setup(22, GPIO.IN)

# Variables for MySQL

db = MySQLdb.connect(host="localhost", user="root",passwd="\*\*\*", db="testing")

cur = db.cursor()

def get\_temp():

#while True:

d0 = GPIO.input(7)

d1 = GPIO.input(11)

d2 = GPIO.input(12)

d3 = GPIO.input(13)

d4 = GPIO.input(15)

d5 = GPIO.input(16)

d6 = GPIO.input(18)

d7 = GPIO.input(22)

value = (d7 << 7) + (d6 << 6) + (d5 << 5) + (d4 << 4) + (d3 << 3) + (d2 << 2) + (d1 << 1) + d0

voltage = (value\*5)/256

temp = voltage\*100

print ("0b%d%d%d%d%d%d%d%d, value=%d, voltage=%f, temp=%f" % \

(d7, d6, d5, d4, d3, d2, d1, d0, value, voltage, temp))

return temp

while True:

temp = get\_temp()

print (temp)

dateWrite = time.strftime("%Y-%m-%d ")

timeWrite = time.strftime("%H:%M:%S")

print (dateWrite,timeWrite)

sql = ("""INSERT INTO tbl\_sensors\_data (sensors\_data\_id,sensors\_temperature\_data,sensors\_data\_date,sensors\_data\_time) VALUES ("",%s,%s,%s)""",(temp,dateWrite,timeWrite))

try:

print ("Writing to database...")

# Execute the SQL command

cur.execute(\*sql)

# Commit your changes in the database

db.commit()

print ("Write Complete")

except:

# Rollback in case there is any error

db.rollback()

print ("Failed writing to database")

time.sleep(10)

**Circuit Diagram Including Modules or Components**

A circuit board

Description automatically generated

1. **Addition design**
2. **Introduction to the architecture**

-IP local host: 10.21.31.182

-Operating system: Noobs

-Software: Python3, Mysql, Php5

Python3

(temperature data)

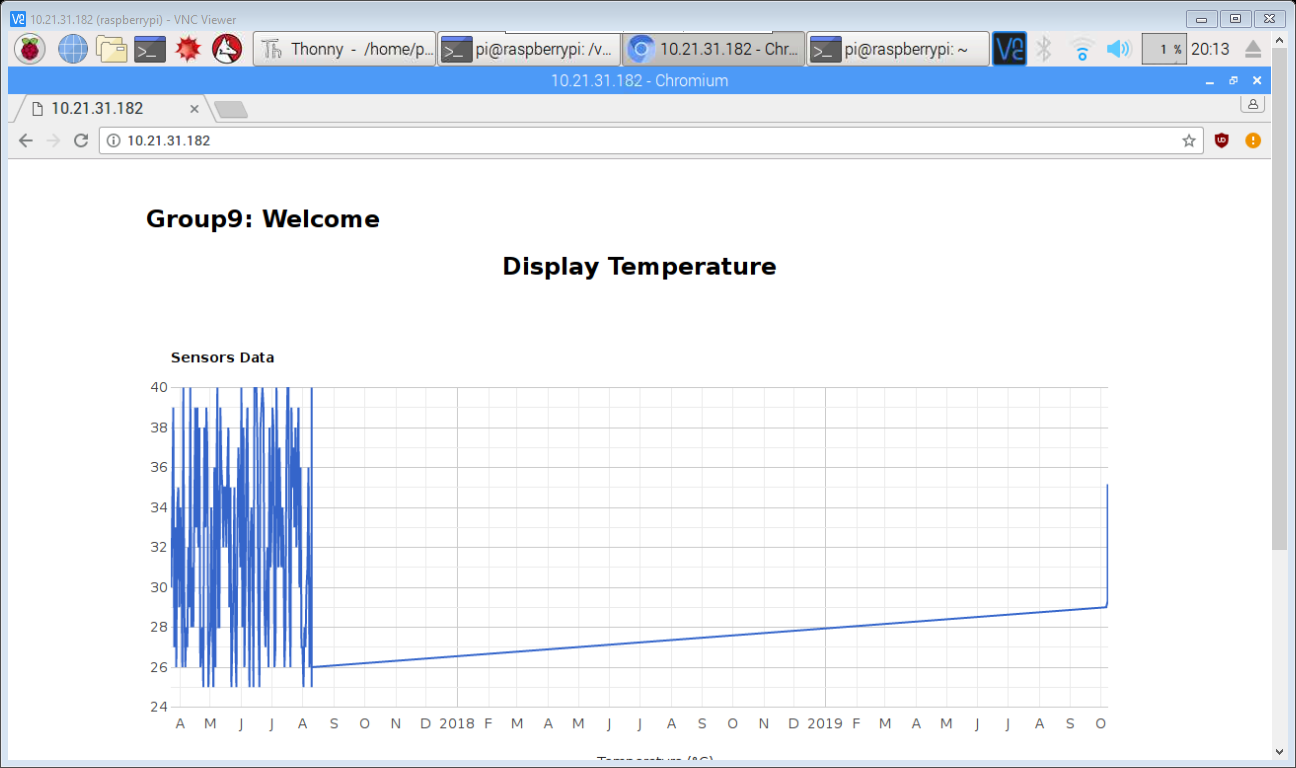
PHP

(show in webpage)

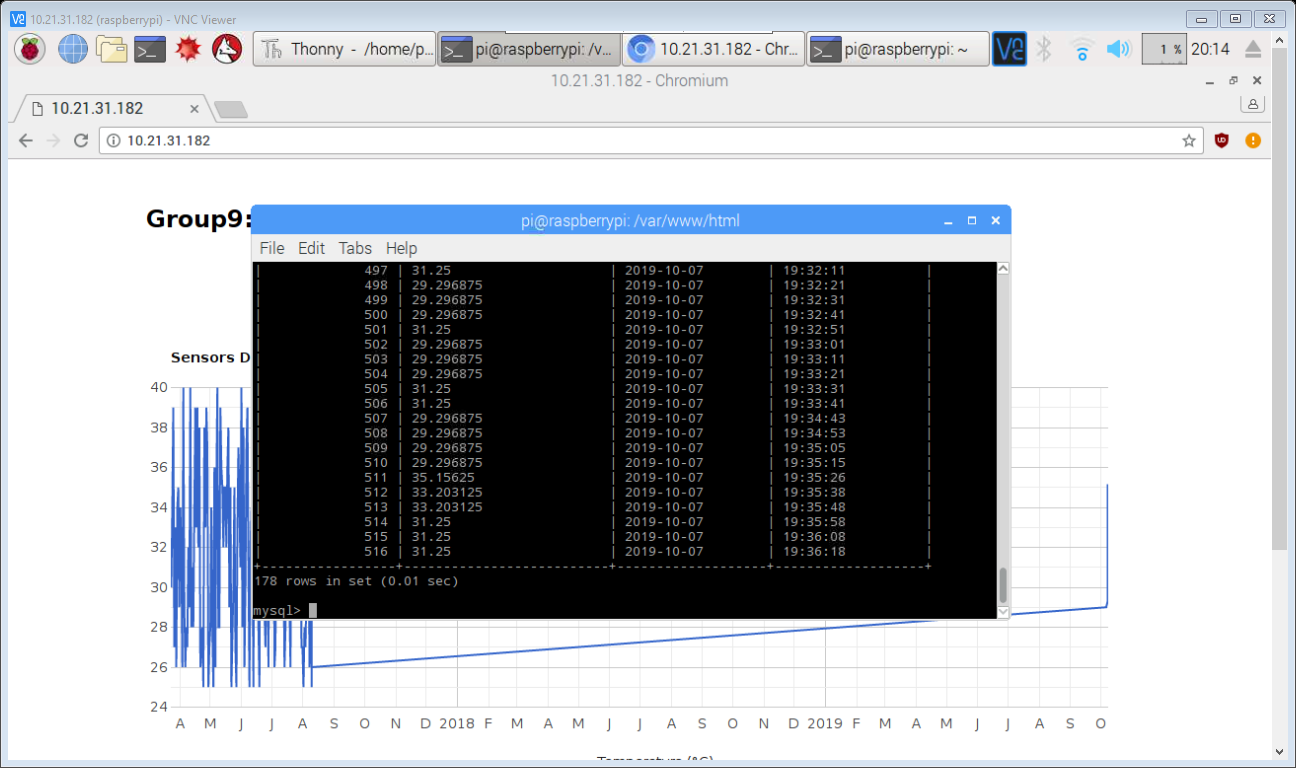
Mysql

(data table)

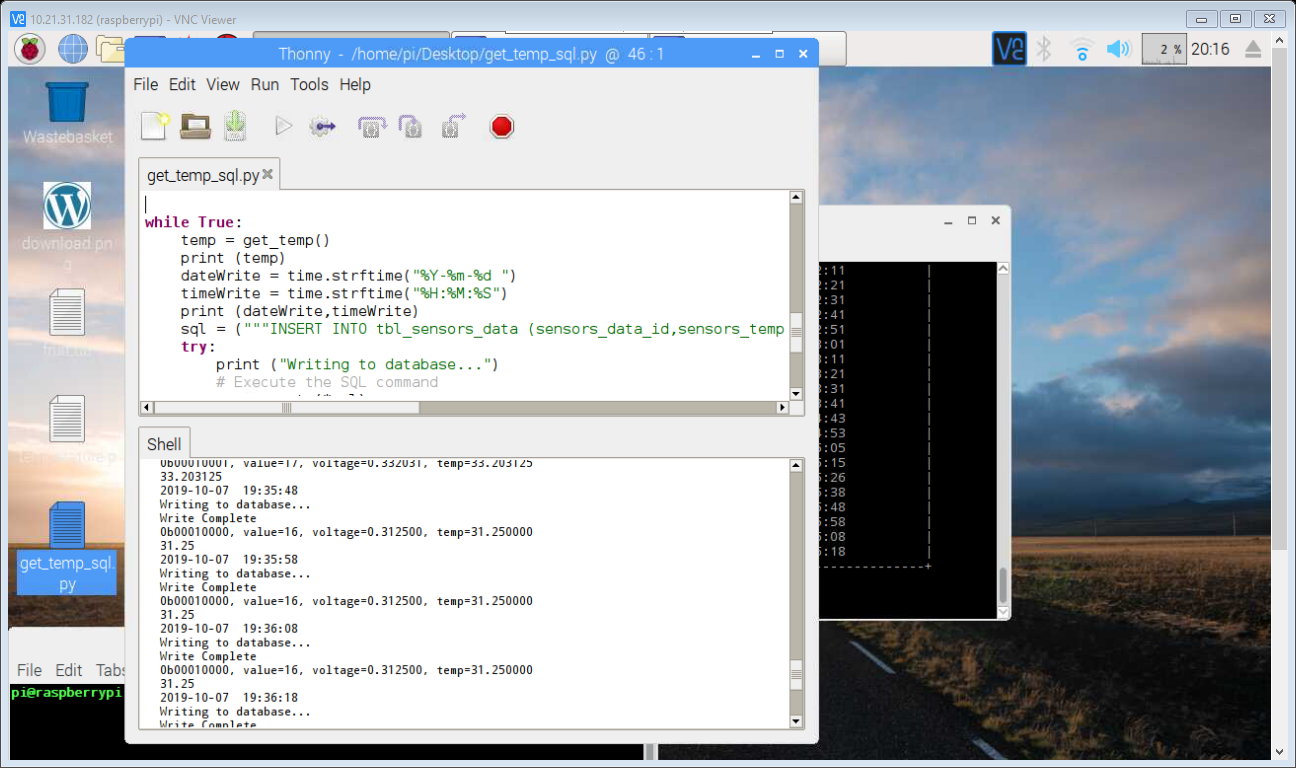
**II. Results of the operation**



**Final result on Webpage**



**Show table in database**



**Sending data from python program**

1. **Collaborations**

Le Minh Hai

Nguyen Thanh Tuan