**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**HỌ VÀ TÊN SINH VIÊN**

**Trần Anh Tuấn  
Trần Văn Cường**

**BÁO CÁO ĐỒ ÁN MÔN HỌC CUỐI KỲ**

**RÚT GỌN BẢNG TRẠNG THÁI   
VÀ HIỂN THỊ LÊN WEB BROWSER**

****

**GIẢNG VIÊN HƯỚNG DẪN**

**ThS. Trương Văn Cương**

**THÀNH PHỐ HỒ CHÍ MINH, 2023**

|  |  |
| --- | --- |
| ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  **TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN** | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **ĐỘC LẬP – TỰ DO – HẠNH PHÚC** |

**CHI TIẾT**

|  |  |
| --- | --- |
| **TÊN ĐỒ ÁN TIẾNG VIỆT: RÚT GỌN BẢNG TRẠNG THÁI VÀ HIỂN THỊ LÊN WEB BROWSER** | |
| **TÊN ĐỒ ÁN TIẾNG ANH: MINIMIZATION STATE TABLE AND DISPLAYING ON WEB BROWSER** | |
| **Giảng viên hướng dẫn:** ThS. Trương Văn Cương, Khoa Kỹ thuật máy tính | |
| **Thời gian thực hiện:** Từ ngày: 15/05/2023 Đến ngày: 31/05/2023 | |
| **Sinh viên thực hiện:** Trần Anh Tuấn, Trần Văn Cường | |
| **Ngôn ngữ lập trình:** Python  **IDE (Integrated Development Environment):** Visual Studio Code | |
| **Chứng nhận của giảng viên**  (chữ ký và họ tên đầy đủ) | **Tp. Hồ Chí Minh, 31/05/2023**  **Sinh viên**  (chữ ký và họ tên đầy đủ) |

**MỤC LỤC**

[**CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI** 1](#_Toc136246281)

[**1.1** **Giới thiệu** 1](#_Toc136246282)

[**1.1.1 Giới thiệu chung** 1](#_Toc136246283)

[**1.1.2 Mô tả hoạt động của hệ thống** 1](#_Toc136246284)

[**1.2 Ngôn ngữ, thư viện, framework sử dụng** 1](#_Toc136246285)

[**1.2.1 Ngôn ngữ lập trình Python** 1](#_Toc136246286)

[**1.2.2 Ngôn ngữ lập trình web HTML, CSS** 1](#_Toc136246287)

[**1.2.3 Thư viện Pandas** 2](#_Toc136246288)

[**1.2.4 Framework Flask** 2](#_Toc136246289)

[**CHƯƠNG 2: LƯU ĐỒ THUẬT TOÁN VÀ CÁC CHƯƠNG TRÌNH** 3](#_Toc136246290)

[**2.1 Lưu đồ thuật toán chương trình chính** 3](#_Toc136246291)

[**2.2 Lưu đồ thuật toán của chương trình phụ** 4](#_Toc136246292)

[**2.3 Chương trình** 6](#_Toc136246293)

[**CHƯƠNG 3: THỰC NGHIỆM** 13](#_Toc136246294)

[**3.1 Điều kiện thực nghiệm** 13](#_Toc136246295)

[**3.2 Kết quả thu được** 13](#_Toc136246296)

[**CHƯƠNG 4: ĐÁNH GIÁ** 14](#_Toc136246297)

[**4.1 Điểm mạnh của hệ thống** 14](#_Toc136246298)

[**4.2 Một số hạn chế còn tồn tại** 14](#_Toc136246299)

[**4.3 Kết luận** 14](#_Toc136246300)

[**TÀI LIỆU THAM KHẢO** 14](#_Toc136246301)

**DANH MỤC HÌNH ẢNH**

[**Hình ảnh 1. 1 Các khối chức năng của hệ thống** 2](#_heading=h.1v1yuxt)

[**Hình ảnh 1. 2 Sơ đồ chân của vi điều khiển 8051** 3](#_heading=h.4f1mdlm)

[**Hình ảnh 1. 3 Sơ đồ chân của LCD 16x2** 4](#_heading=h.2u6wntf)

[**Hình ảnh 1. 4 Mã lệnh giao tiếp với LCD16x2** 6](#_heading=h.19c6y18)

[**Hình ảnh 1. 5 Sơ đồ chân ADC0804** 7](#_heading=h.3tbugp1)

[**Hình ảnh 2. 2 Lưu đồ thuật toán chương trình chính** 11](#_heading=h.lnxbz9)

[**Hình ảnh 2. 3 Lưu đồ thuật toán chương trình phụ** 12](#_heading=h.1ksv4uv)

# **CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI**

## **Giới thiệu**

### **1.1.1 Giới thiệu chung**

Chỉ trong vỏn vẹn vài thập kỉ vừa qua, nhân loại chúng ta mặc dù đã trải qua biết bao là bước tiến vĩ đại và mạnh mẽ trong cả lĩnh vực phần cứng và phần mềm, tuy nhiên việc sử dụng FSM vẫn là một phương pháp quan trọng để mô hình hóa hệ thống. Gắn liền với FSM, việc quản lý số lượng trạng thái lại là một thách thức không nhỏ đối với các nhà phát triển bởi có thể có nhiều FSM cùng chung chức năng nhưng khác nhau về số lượng trạng thái.

→ Việc tìm ra FSM có chức năng tương tự với số lượng trạng thái là nhỏ nhất (rút gọn trạng thái) cực kỳ quan trọng khi nó mang lại hiệu suất cao khi thiết kế phần cứng cũng như thời gian xử lý của hệ thống sẽ được tối ưu.

### **1.1.2 Mô tả hoạt động của hệ thống**

- Ban đầu, chúng ta sẽ sử dụng ngôn ngữ lập trình Python và thư viện Pandas để thu thập data của bảng trạng thái cần rút gọn từ file excel.

- Sau đó, thực hiện rút gọn trạng thái dựa trên data đã thu thập

- Cuối cùng, lấy data sau khi rút gọn hiển thị lên web browser bằng cách sử dụng HTML, CSS và framework Flask.

## **1.2 Ngôn ngữ, thư viện, framework sử dụng**

### **1.2.1 Ngôn ngữ lập trình Python**

- Python là ngôn ngữ lập trình cấp cao mạnh mẽ, dễ học với cú pháp đơn giản, ngắn gọn và có thể ứng dụng ở nhiều lĩnh vực khác nhau như: khoa học dữ liệu, AI, phát triển phần mềm nhúng, …

- Ngoài ra, Python còn được hỗ trợ bởi nhiều thư viện mạnh mẽ như Pandas, Openpyxl, Matplotlib, …

- Các lý do nhóm lựa chọn Python thay vì C++ :

+ Hỗ trợ đọc file excel nhanh gọn.

+ Cú pháp ngắn gọn, dễ đọc hiểu, bảo trì code.

+ Là một cách để học thêm ngôn ngữ mới thông qua dự án thực tế.

### **1.2.2 Ngôn ngữ lập trình web HTML, CSS**

- HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản.

- Không phải là ngôn ngữ lập trình.

- HTML có tác dụng tạo bố cục và định dạng trang web.

- CSS (Cascading Style Sheets) định nghĩa về cách hiển thị của một tài liệu HTML.

- CSS đặc biệt hữu ích trong việc thiết kế Web. Nó giúp cho người thiết kế dễ dàng đặt các style đã được thiết kế lên bất kì page nào của website một cách nhanh chóng và đồng bộ.

### **1.2.3 Thư viện Pandas**

- Pandas là một thư viện mã nguồn mở được phát triển dựa trên ngôn ngữ Python bởi Wes Mckinney năm 2008.

- Pandas thực hiện phân tích và thao tác dữ liệu; bất kỳ loại xử lý, phân tích, lọc và tổng hợp dữ liệu.

- Những phần sử dụng trong đồ án:

+ Kiểu dữ liệu DataFrame: là 1 kiểu dữ liệu như mảng 2 chiều hoặc giống như bảng gồm nhiều hàng và cột.

* Hàm dropna() : giúp loại bỏ các dòng chứa giá trị NULL.
* Hàm rename() : giúp đổi tên index hàng hoặc tên cột.

+ Hàm read\_excel() : đọc data từ file excel và trả về dưới dạng một dataframe.

### **1.2.4 Framework Flask**

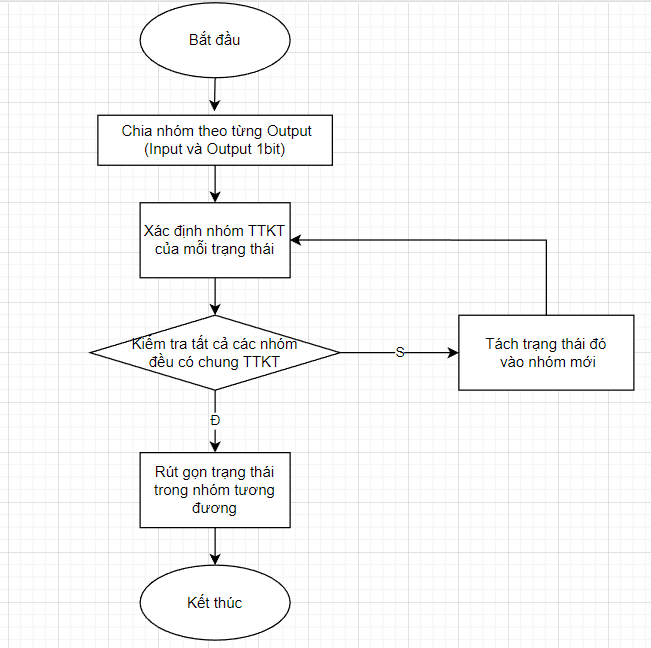
- Flask là một framework web nhẹ, linh hoạt và dễ sử dụng được viết bằng Python, cung cấp cho người dùng một cách tiếp cận tối giản hóa việc phát triển ứng dụng web.

- Nó được sử dụng để xây dựng các ứng dụng web, API và các dịch vụ web khác.

- Mục đích sử dụng trong đồ án: dùng để truyền data từ Python sang cho HTML để hiển thị lên web browser.

# **CHƯƠNG 2: LƯU ĐỒ THUẬT TOÁN VÀ CÁC CHƯƠNG TRÌNH**

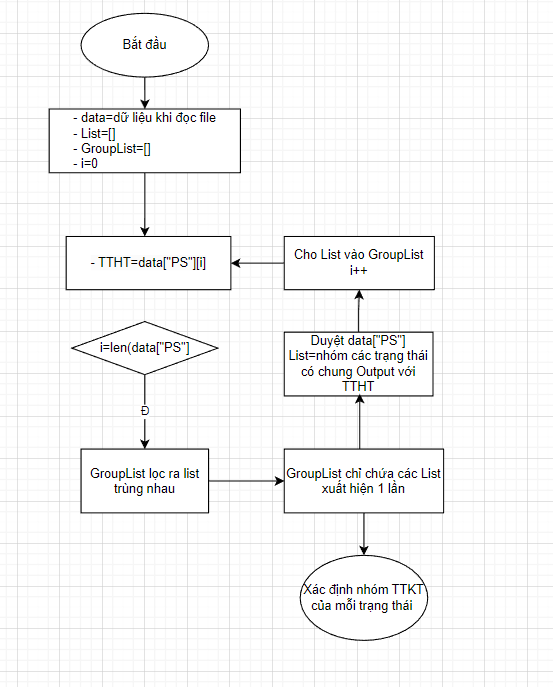
## **2.1 Lưu đồ thuật toán chương trình chính**



***Hình ảnh 2.1 Lưu đồ thuật toán chương trình chính***

## **2.2 Lưu đồ thuật toán của chương trình phụ**

* Lưu đồ thuật toán chương trình chia nhóm theo từng Output:



***Hình ảnh 2.2.1 Lưu đồ thuật toán chương trình chia nhóm theo từng Output***

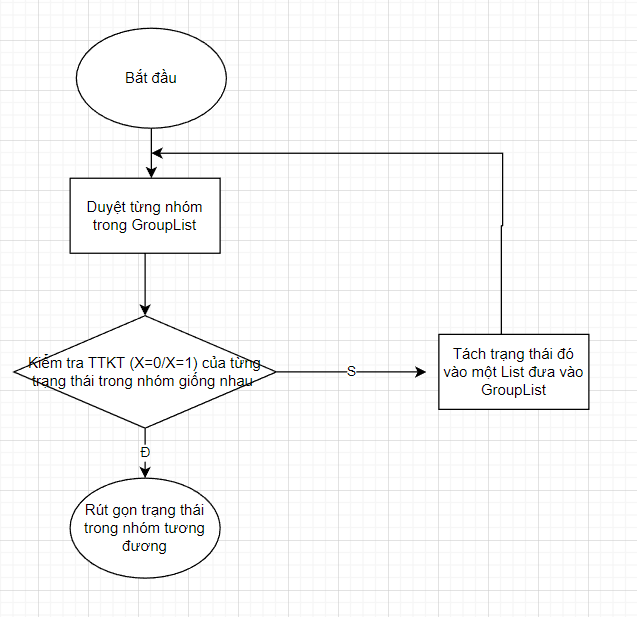
* Lưu đồ thuật toán chương trình xác định nhóm TTKT :

A picture containing text, diagram, line, screenshot

Description automatically generated

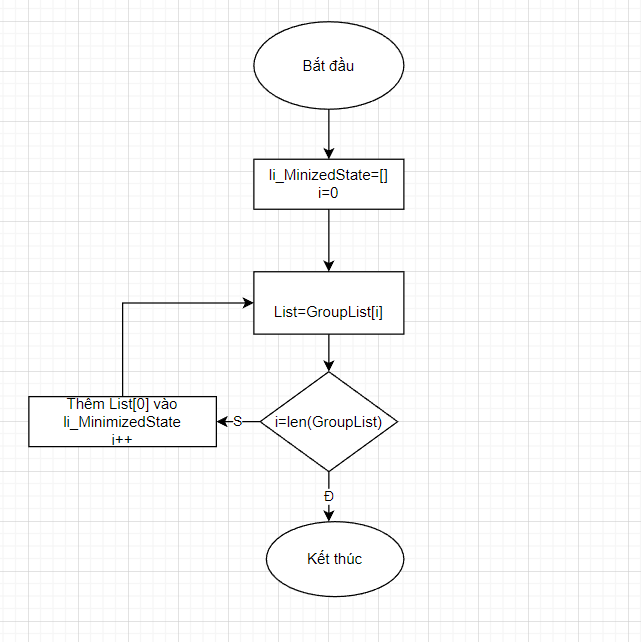
**Hình ảnh 2.2.2 Lưu đồ thuật toán chương trình xác nhóm TTKT**

- Lưu đồ thuật toán chương trình kiểm tra điều kiện và tách nhóm:



**Hình ảnh 2.2.3 Lưu đồ thuật toán chương trình kiểm tra điều kiện và tách nhóm**

- Lưu đồ thuật toán chương trình rút gọn trạng thái trong nhóm tương đương:



**Hình ảnh 2.2.4 Lưu đồ thuật toán chương trình rút gọn trạng thái trong nhóm tương đương**

## **2.3 Chương trình**

|  |
| --- |
| # import library :  import pandas as pd  from flask import Flask,render\_template  import time  start\_time = time.time()  app=Flask(\_\_name\_\_)  # read file by pandas : data is Dataframe  data=pd.read\_excel( r'autoTestCase/autoTest1*.*xlsx')  # remove  any rows in Dataframe contains NaN values  data.dropna(*inplace*=True)  # rename colums in DataFrame to new short name  data.rename( *columns*={'PRESENT STATE':'PS' }, *inplace*=True )  data.rename( *columns*={'NEXT STATE':'NS0' }, *inplace*=True )  data.rename( *columns*={'Unnamed: 2':'NS1'}, *inplace*=True )  data.rename( *columns*={'OUTPUT':'O0' }, *inplace*=True )  data.rename( *columns*={'Unnamed: 4':'O1'}, *inplace*=True)  # PART 1: Divide Group by common output  # Create List contain lists of state having common output  def devideGroupByOutput():      groupList=[]      for i in range(1,len(data["PS"]) + 1):          List=[]          for j in range(1,len(data["PS"]) + 1):              if data['O0'][i]==data['O0'][j] and data['O1'][i]== data['O1'][j]:                  List.append(data["PS"][j])          groupList.append(List)      return groupList  # Minimize GroupList include lists include 1 time  def minimizeGroupList(*groupList*):      for *list* in *groupList*:          while *groupList*.count(*list*) >1:  *groupList*.remove(*list*)  #Part 2:  # create Dictionary of each state when X=0  # create Dictionary of each state when X=1  def createDictNextState(*dict\_NS0*,*dict\_NS1*):      for i in range(1,len(data["PS"]) + 1):          name=data["PS"][i]          valueNS0=[]          valueNS1=[]          for lst in groupList:              if data["NS0"][i] in lst:                  valueNS0=lst              if data["NS1"][i] in lst:                  valueNS1=lst  *dict\_NS0*[name]=valueNS0  *dict\_NS1*[name]=valueNS1  def checkListNoDifferent(*lists*):      for i in *lists*:          for j in *lists*:              if i !=j :                  return False      return True  #PART 3: Check Condition and divide new list in groupList  def findListAppearMinTime(*lists*):      min= *lists*.count(*lists*[0])      for i in *lists*:          if *lists*.count(i) <= min:              min=*lists*.count(i)      return min  def findListDifferent(*parentlist*):      if len(*parentlist*)== 1 or checkListNoDifferent(*parentlist*) == True:          return None      for lst in *parentlist*:          if *parentlist*.count(lst) == 1:              return lst          elif *parentlist*.count(lst) == findListAppearMinTime(*parentlist*):              return lst      return None  def findStringDifferent(*parentlist*,*dictNS*,*list*):      li\_string=[]      if findListDifferent(*parentlist*) is None:          return None      else:          for i in *list*:              if *dictNS*[i] is findListDifferent(*parentlist*):                  li\_string.append(i)      return li\_string  def findListNextStateInGroup(*lst*,*dict\_NS*):          List\_NS0=[]          for i in *lst*:              List\_NS0.append(*dict\_NS*[i])          return List\_NS0  def updateDictNextState():      for i in range(1,len(data["PS"]) + 1):          for lst in groupList:              if data["NS0"][i] in lst:                  dict\_NS0[data["PS"][i]]=lst              if data["NS1"][i] in lst:                  dict\_NS1[data["PS"][i]]=lst  def findListRest(*lst*,*List*):      li=[]      for i in *List*:          if len(*lst*)==1:              if i != *lst*[0]:                  li.append(i)          elif i not in *lst* and *List* != None:              li.append(i)      return li  def divideGroupbyNextState():      for lst in groupList:          #check states in lst whether their nextStates are common          stateDifferentInput0=findStringDifferent(findListNextStateInGroup(lst,dict\_NS0),dict\_NS0,lst)          stateDifferentInput1=findStringDifferent(findListNextStateInGroup(lst,dict\_NS1),dict\_NS1,lst)          # if Diffrent, add new list in groupList          if stateDifferentInput0 != None or stateDifferentInput1 !=None:              if stateDifferentInput0 != None:                  lstRest= findListRest(stateDifferentInput0,lst)                  groupList.remove(lst)                  groupList.append(stateDifferentInput0)                  groupList.append(lstRest)              elif stateDifferentInput1 != None:                  lstRest=findListRest(stateDifferentInput1,lst)                  groupList.remove(lst)                  groupList.append(stateDifferentInput1)                  groupList.append(lstRest)      # Updated in Dict\_NS0,DictNS\_1 when checking of states      updateDictNextState()  # Part 3: Loop and perform recursion in groupList to create entire equivalent list  def minimizeState():      for lst in groupList:          stateDifferentInput0=findStringDifferent(findListNextStateInGroup(lst,dict\_NS0),dict\_NS0,lst)          stateDifferentInput1=findStringDifferent(findListNextStateInGroup(lst,dict\_NS1),dict\_NS1,lst)          if stateDifferentInput0 != None or stateDifferentInput1 != None:              divideGroupbyNextState()              minimizeState()  # Part 4: Minimize list in groupList by get 1 value in list  def createDataMinimize():  *dict*={}      li=[]      # create List of Present State Minimize:      li\_PSMinimize=[]      for lst in groupList:          li.append(lst[0])      for i in dict\_NS0:          for j in li:              if i == j:                  li\_PSMinimize.append(i)      li\_Output0=[]      for state in li\_PSMinimize:          for i in range(1,len(data["PS"]) + 1):               if data["PS"][i] == state:                  li\_Output0.append(data["O0"][i])      li\_Output1=[]      for state in li\_PSMinimize:          for i in range(1,len(data["PS"]) + 1):               if data["PS"][i] == state:                  li\_Output1.append(data["O1"][i])      li\_NextState0=[]      for state in li\_PSMinimize:          li\_NextState0.append(dict\_NS0[state][0])      li\_NextState1=[]      for state in li\_PSMinimize:          li\_NextState1.append(dict\_NS1[state][0])      dict["PS"]=li\_PSMinimize      dict["NS0"]=li\_NextState0      dict["NS1"]=li\_NextState1      dict["O0"]=li\_Output0      dict["O1"]=li\_Output1      return *dict*  #MAIN  groupList=[]  # Step 1:  groupList = devideGroupByOutput()  minimizeGroupList(groupList)  #Step 2:  dict\_NS0={}  dict\_NS1={}  createDictNextState(dict\_NS0,dict\_NS1)  # step 3:  minimizeState()  minimizeGroupList(groupList)  #step 4:  dataAfter=createDataMinimize()  # Put data has been solved to display in Web browser by Flask  len\_DataAfter=len(groupList)  len\_Data=len(data)  li\_LenMiniMize=[i for i in range(0,len(dataAfter["PS"]))]  li\_LenData=[i for i in range(1,len(data["PS"]) + 1)]  # Using library "time" to caculate total\_time  end\_time = time.time()  total\_time = end\_time - start\_time  print("Thời gian chạy: ", total\_time, " giây")  # Using Flask to return template to file index.html  @app.route('/')  def index():      return render\_template("index2.html",*dataAfter*=dataAfter,*len\_DataAfter*=len\_DataAfter,*data*=data,*len\_Data*=len\_Data,*li\_LenMiniMize*=li\_LenMiniMize,*li\_LenData*=li\_LenData )  if \_\_name\_\_ == '\_\_main\_\_':      app.run(*debug*=True) |

# **CHƯƠNG 3: THỰC NGHIỆM VÀ ĐÁNH GIÁ**

## **3.1 Kết quả thu được**

-

# **CHƯƠNG 4: ĐÁNH GIÁ**

## **4.1 Điểm mạnh của hệ thống**

- Trong môi trường mô phỏng nhiệt độ từ 0 – 150 độ C, hệ thống đáp ứng rất tốt về khả năng hiển thị nhiệt độ với sai số chấp nhận được (1 độ C) và thời gian đáp ứng rất nhanh gần như không có độ trễ.

- Phát cảnh báo chính xác và nhanh chóng khi nhiệt độ vừa chạm mức 50 độ C.

## **4.2 Một số hạn chế còn tồn tại**

- Hệ thống vẫn chưa hiển thị được nhiệt độ âm và chưa giảm được sai số xuống dưới 1 độ C.

- Giải thích sai số 1 độ C của hệ thống:

+ Công thức chuyển đổi giá trị Vout sang nhiệt độ t hiển thị lên LCD:

+ Sai số 1 độ C là vì :

* Sai số của cảm biến nhiệt độ LM35 là 0.5 độ C.
* Sai số khi làm tròn giá trị t từ kiểu float sang kiểu int trong ngôn ngữ lập trình C.

## **4.3 Kết luận**

- Vì đây là hệ thống được thực hiện trên phần mềm mô phỏng nên so với thực tế sẽ còn nhiều sự khác biệt.

- Hướng phát triển: cải tiến hệ thống giúp hệ thống hiển thị được nhiệt độ âm, đồng thời kết hợp thêm cảm biến khói để phát triển hệ thống báo cháy và chữa cháy.

# **TÀI LIỆU THAM KHẢO**

[1] Muhammad Ali Mazidi & Janice Gillispie Mazidi, *The 8051 Microcontroller and Embedded Systems: Using Assembly and C, Second Edition*, Pearson Prentice Hall, 2006.

[2] Tống Văn On, *Họ vi điều khiển 8051*, Nhà xuất bản Lao động – Xã hội.