

Bài 8: Python giao tiếp Arduino

I. MỤC ĐÍCH- YÊU CẦU

- Ngôn ngữ lập trình Python cho hệ thống nhúng
- Giao tiếp máy tính (hoặc máy tính nhúng) với Arduino dùng python
- Dùng Visual Studio Code để viết chương trình Python

II. GIỚI THIỆU

Bài thực hành cho phép giao tiếp với các board Arduino qua giao tiếp nối tiếp UART0 trên board Arduino. UART0 cũng là giao tiếp để giao tiếp với bootloader trên board arduino khi nạp chương trình và là cổng để gửi và nhận thông tin trên Terminal. Sinh viên cần thao tác chính xác để tránh vấn đề tranh chấp cổng giao tiếp này. Trên máy tính sinh viên cần xác định số cổng COM giao tiếp với Arduino để thay đổi cho phù hợp.

III. NỘI DUNG THỰC HÀNH

1. Giao tiếp UART dùng thư viện pyserial

Trong Vscode, trên menu Terminal chọn new Terminal. Ở giao diện dòng lệnh để cài đặt thư viện giao tiếp nối tiếp ta nhập vào lệnh: *pip install pyserial*

```
PS D:\HeThong\hung\code\python> pip install pyserial
Collecting pyserial
  Obtaining dependency information for pyserial from https://files.pythonhosted.org/packages/07/bc/587a445451b253b285629263eb51c2d8e9bcea4fc97826266d186f96f558/pyserial-3.5-py2.py3-none-any.whl.metadata (1.6 kB)
Using cached pyserial-3.5-py2.py3-none-any.whl (90 kB)
Installing collected packages: pyserial
Successfully installed pyserial-3.5
```

Thư viện pyserial là thư viện dùng chính để giao tiếp với board Arduino thông qua giao tiếp nối tiếp UART. Thông tin chi tiết về thư viện có thể tham khảo theo link:

<https://pyserial.readthedocs.io/en/latest/>

Các thư viện cài đặt thêm sẽ được cài vào ở: [Python\Lib\site-packages](#).

Đoạn chương trình python sau sẽ tạo kênh nối tiếp và gửi 1 chuỗi xuống kết nối này

```
1 import serial
2 def main():
3     ser=serial.Serial('COM3') #mở port giao tiếp COM
4     print(ser.name)
5     ser.write(b'Hello') #gửi một chuỗi qua port nối tiếp
6     ser.close() #đóng kết nối
7
8 if __name__=='__main__':
9     main()
```

Trên Arduino bạn hãy viết chương trình nhận dữ liệu từ cổng nối tiếp Serial, nếu chuỗi nhận được là “Hello” thì sáng đèn Led đơn nối với chân D13.

Chạy thử nghiệm giao tiếp này. Chương trình python chỉ mở kết nối, gửi một chuỗi và kết thúc chương trình do đó đòi hỏi Arduino phải chạy trước để chuẩn bị nhận chuỗi.

Chú ý: khi giao tiếp không được mở Serial monitor trên Arduino IDE, sẽ chiếm kết nối Serial0 làm chương trình Python không mở kết nối được.

Dựa vào chương trình ở phần trên và các bài thực hành Arduino trước đây, bạn hãy viết chương trình giao tiếp với Python để điều khiển bật tắt 16 Led đơn để tạo hiệu ứng 16 led sáng dần và tắt dần.

Viết chương trình Python giao tiếp Arduino đọc giá trị nhiệt độ từ cảm biến nhiệt độ DHT11, giá trị nhiệt độ hiển thị trên terminal của Python.

2. Giao tiếp Arduino dùng pyFirmata

Thư viện pyFirmata2 là thư viện triển khai giao thức Firmata trên python cho phép giao tiếp điều khiển Arduino trực tiếp từ máy tính. Trên terminal của VScode cài thư viện pyFirmata2 bằng lệnh `pip install pyfirmata2`.

Trên Arduino mở chương trình triển khai giao thức Firmata chuẩn ở File→Examples→Firmata→StandardFirmata. Biên dịch và nạp chương trình vào Arduino.

Trên Python viết chương trình điều khiển sáng tắt Led D13 trên Arduino như sau:

```
1  ∨ from pyfirmata2 import Arduino
2  from time import sleep
3  board=Arduino('COM1') #mở kết nối với Arduino
4  ∨ while True:
5      board.digital[13].write(1) #ghi mức 1 ra chân 13
6      sleep(0.5) #delay 0.5s
7      board.digital[13].write(0) # ghi mức 0 ra chân 13
8      sleep(0.5)
```

Chạy thử chương trình python, quan sát trạng thái Led trên chân D13, đây là vòng lặp vô hạn nên chương trình Python sẽ không kết thúc, để kết thúc ta nhấn tổ hợp phím Ctrl+C.

3. Chương trình python sau để đọc trạng thái nút nhấn:

```
1 import pyfirmata2
2 PORT = pyfirmata2.Arduino.AUTODETECT
3 # Hàm Callback được gọi khi có chuyển trạng thái trên chân nút nhấn
4 def pinCallback(value):
5     if value:
6         print("Button released")
7     else:
8         print("Button pressed")
9 board = pyfirmata2.Arduino(PORT)
10 # định thời gian lấy mẫu mặc định 19ms
11 board.samplingOn()
12 # thiết lập chân D13 ở chế độ input pullup "u"
13 digital_0 = board.get_pin('d:6:u')
14 # chỉ đến hàm callback
15 digital_0.register_callback(pinCallback)
16 # Bật chế độ cho phép callback
17 digital_0.enable_reporting()
18 input("Nhấn enter để kết thúc chương trình")
19 board.exit()
```

Chạy thử chương trình python trên, quan sát thông tin xuất ra màn hình Terminal khi nhấn và nhả nút.

Dựa vào chương trình trên bạn hãy viết chương trình đảo trạng thái Led D13 sau mỗi lần nhấn nút nhấn.

4. Chương trình Python sau điều khiển góc quay của động cơ RC Servo

```
1 import pyfirmata2
2 PORT = pyfirmata2.Arduino.AUTODETECT #tự động phát hiện port giao tiếp với Arduino
3 board = pyfirmata2.Arduino(PORT)
4 # thiết lập chân điều khiển RC Servo
5 servo_6 = board.get_pin('d:6:s') #chân D6 phát xung điều khiển servo
6 v = int(input("Nhập vào góc quay mong muốn cho servo (góc quay <180 độ) ")) #doc gia tri va chuyen sang so
7 servo_6.write(v)
8 input("Nhấn phím Enter để kết thúc chương trình")
9 # Dòng kết nối
10 board.exit()
```

Chạy thử chương trình trên.

Sửa lại chương trình trên để liên tục điều khiển góc quay động cơ Servo, chương trình chỉ kết thúc khi người dùng nhập vào số -1.