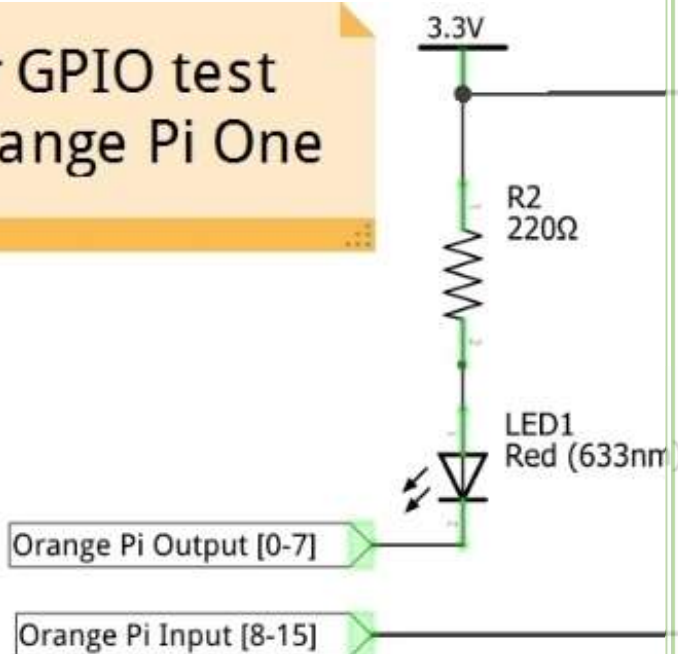


2016

# การเขียนโปรแกรมพื้นฐาน 101

for GPIO test  
Orange Pi One



เริ่มต้นกับ ภาษาไพทอน

กับ สวิตช์ และหลอดไฟ

10/8/2016

## Table of Contents

---

|   |    |
|---|----|
| การเตรียมตัวเบื้องต้น .....                       | 2  |
| แนวคิดในการพัฒนาโปรแกรม .....                     | 7  |
| การพัฒนาโปรแกรมแบบสเตต-แมชชีน(State-Machine)..... | 7  |
| การพัฒนาโปรแกรมด้วยโฟลว์ชาร์ต .....               | 9  |
| สรุปตัวอย่างคำสั่งต่างๆในภาษา Python .....        | 13 |
| การประกาศตัวแปร ในรูปแบบต่างๆ.....                | 13 |
| การแทรกคำอธิบาย(Comment) .....                    | 13 |
| ตัวแปรประเภทข้อความ (string) .....                | 13 |
| ตัวแปรประเภทจำนวนเต็ม (Integer) .....             | 14 |
| ตัวแปรประเภทจำนวนทศนิยม (Float) .....             | 15 |
| ตัวแปรประเภทชุดลำดับ(Lists).....                  | 15 |
| เชิงอรรถ .....                                    | 17 |
| อ้างอิง.....                                      | 17 |

## การเขียนโปรแกรมพื้นฐาน 101

### การเตรียมตัวเบื้องต้น

โปรแกรมแรกๆ ที่โปรแกรมเมอร์มักจะเริ่มต้นทำกันคือ การเขียนคำสั่งให้ แสดงผลคำว่า “Hello World” ลงบนคอนโซลแสดงผล สำหรับภาษา Python นั้น ใช้เพียงคำสั่งสั้นๆ ง่ายๆ ว่า

```
print “Hello World”
```

เพียงเท่านี้ เราก็จะสามารถแสดงผลคำว่า Hello World<sup>[1]</sup> ได้แล้ว

แต่... จะพิมพ์ที่ไหนล่ะ? ยังไงล่ะ? พิมพ์แล้ว จะเห็นผลเลยหรือ? ต้องทำอะไรก่อนไหม?

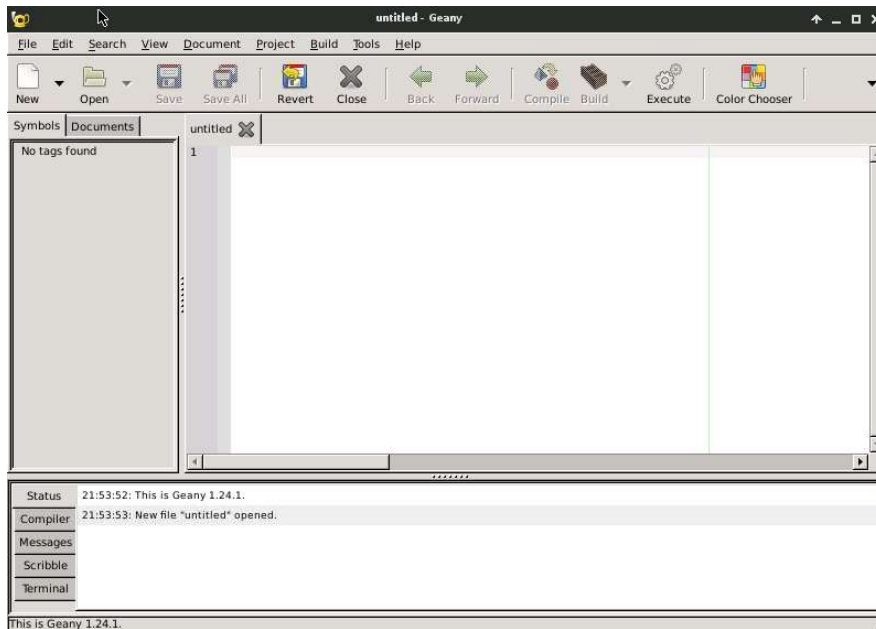
นี่แหละครับ จุดประสงค์ของคำว่า “Hello World” คือ ถ้าไม่สามารถทักทายกันได้ แสดงว่าเรายังใช้งานมันไม่ได้นั่นเอง

เรามาเริ่มกันเลย โดยเลือกที่เมนู *Application -> Development -> Geany*

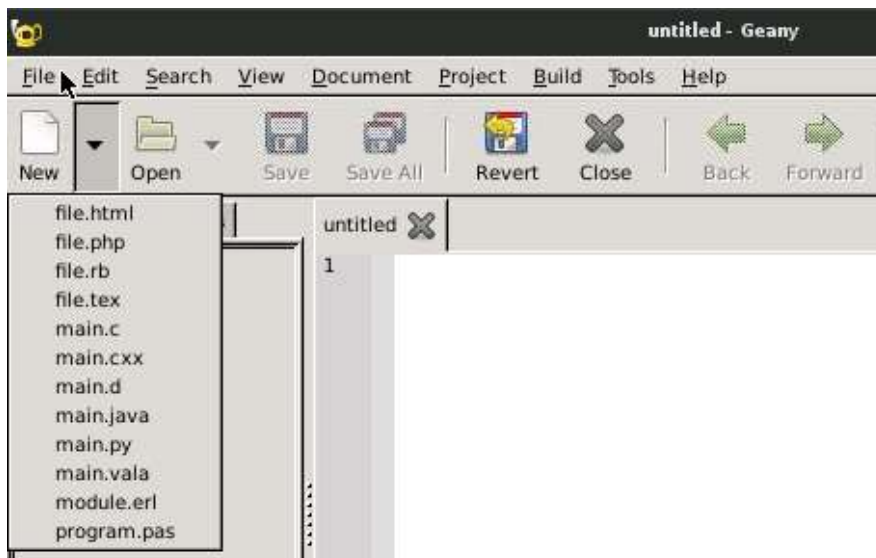


## การเขียนโปรแกรมพื้นฐาน 101

Geany คือ สิ่งที่เราเรียกว่า IDE (Integrated Development Environment) เป็นเครื่องมือ ที่ถูกสร้างขึ้นมา เพื่อให้เราสามารถพัฒนาโปรแกรมได้ง่ายมากขึ้น ยุ่งยากน้อยลง (แม้ว่าจะแคในบางครั้งก็ตาม) เมื่อเราเปิด ขึ้นมาก็จะพบหน้าต่างแบบนี้

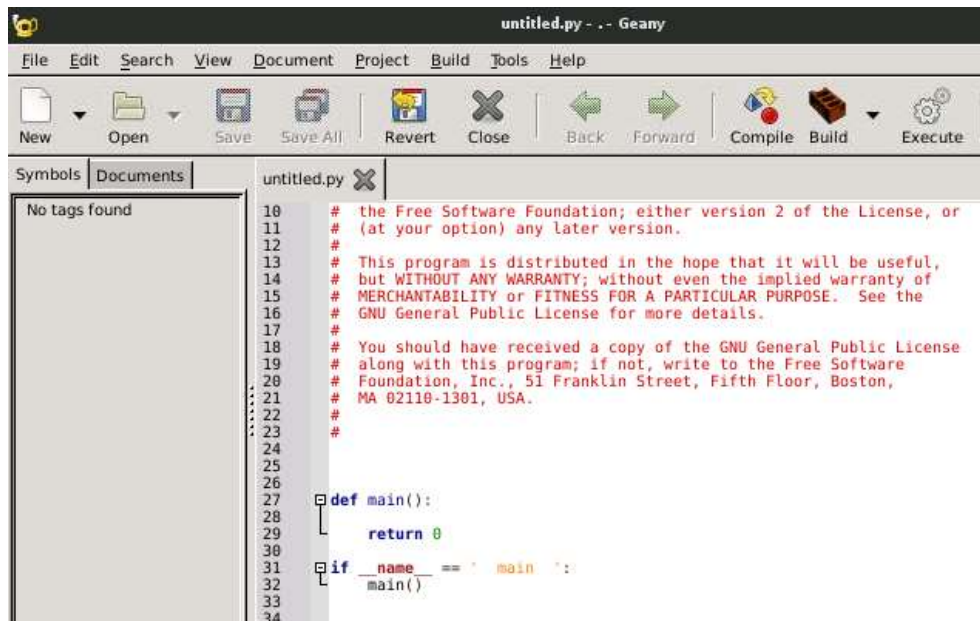


เมื่อเราจะสร้างโค้ด ก็เริ่มด้วยไปที่เมนู New -> main.py



## การเขียนโปรแกรมพื้นฐาน 101

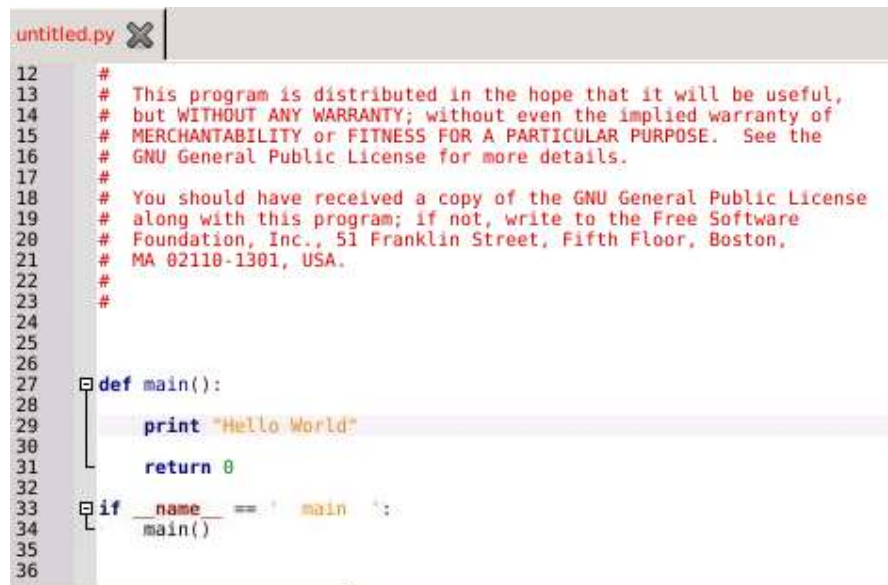
จากนั้นเราก็จะได้ รูปแบบโค้ดมาตรฐานที่ Geany สร้างมา สำหรับให้เราเขียนโปรแกรมได้ง่ายขึ้น



```
10 # the Free Software Foundation; either version 2 of the License, or
11 # (at your option) any later version.
12 #
13 # This program is distributed in the hope that it will be useful,
14 # but WITHOUT ANY WARRANTY; without even the implied warranty of
15 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16 # GNU General Public License for more details.
17 #
18 # You should have received a copy of the GNU General Public License
19 # along with this program; if not, write to the Free Software
20 # Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
21 # MA 02110-1301, USA.
22 #
23 #
24 #
25 #
26 #
27 def main():
28     return 0
29
30 if __name__ == '__main__':
31     main()
32
33
34
```

เมื่อเราได้โค้ดเริ่มต้นมาแล้ว และเป้าหมายอันยิ่งใหญ่ของเราคือ ทักทายกับคอมพิวเตอร์ด้วยคำว่า “Hello World”

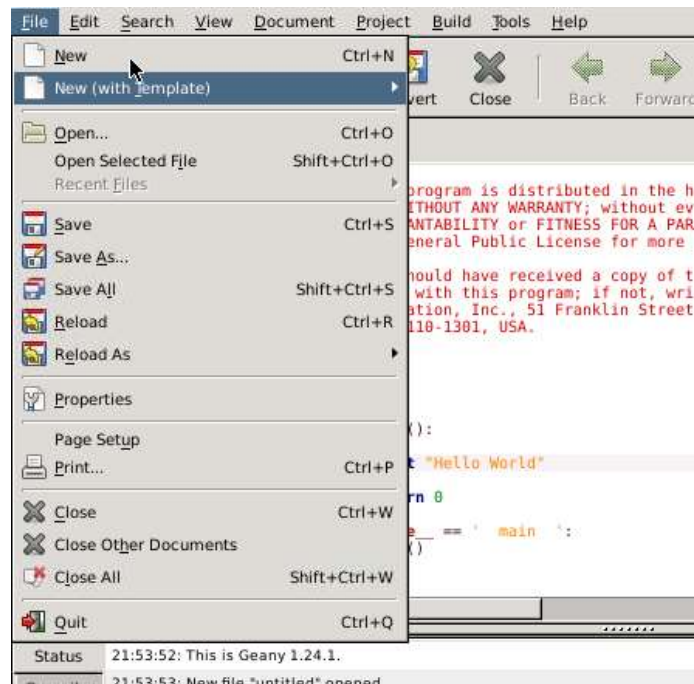
เราสามารถเพิ่มโค้ด “Hello World” ของเราได้ในบรรทัดหลัง def main(): และก่อน return 0 ดังตัวอย่าง



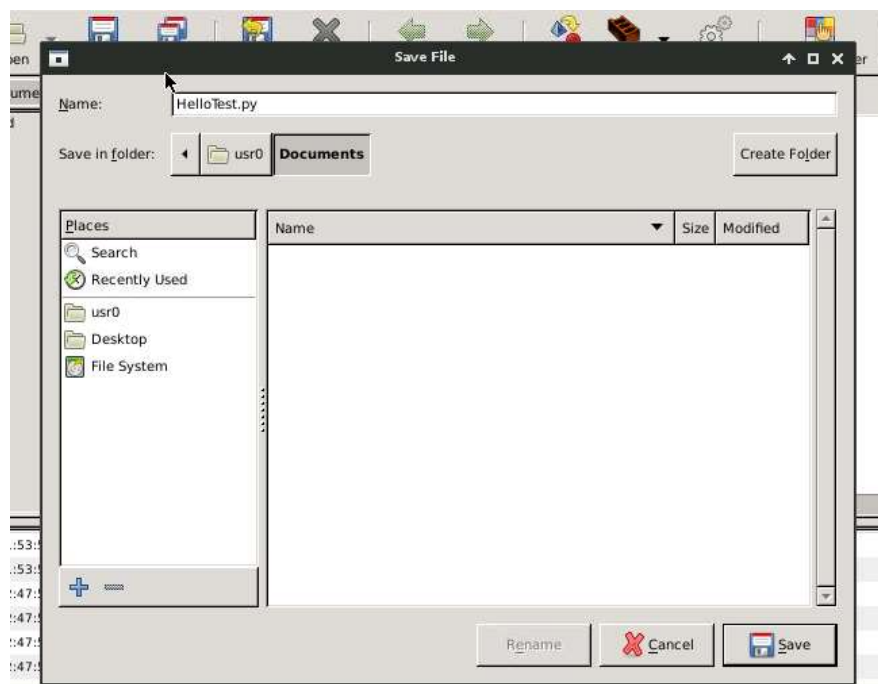
```
12 #
13 # This program is distributed in the hope that it will be useful,
14 # but WITHOUT ANY WARRANTY; without even the implied warranty of
15 # MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
16 # GNU General Public License for more details.
17 #
18 # You should have received a copy of the GNU General Public License
19 # along with this program; if not, write to the Free Software
20 # Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston,
21 # MA 02110-1301, USA.
22 #
23 #
24 #
25 #
26 #
27 def main():
28     print "Hello World"
29     return 0
30
31 if __name__ == '__main__':
32     main()
33
34
35
36
```

## การเขียนโปรแกรมพื้นฐาน 101

แล้วก็ทำการเซฟไฟล์ของเรา โดยไปที่เมนู File -> Save As



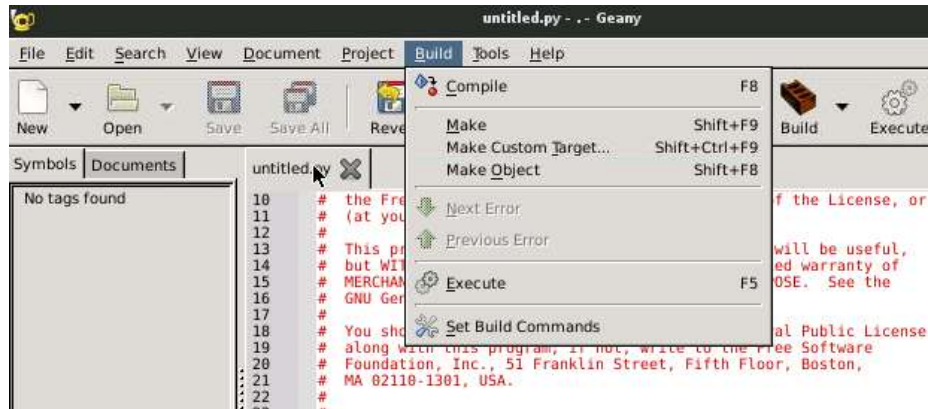
จากนั้นก็เปลี่ยนชื่อไฟล์ และเลือกที่เก็บไฟล์ได้ในหน้านี้ แล้วกด Save



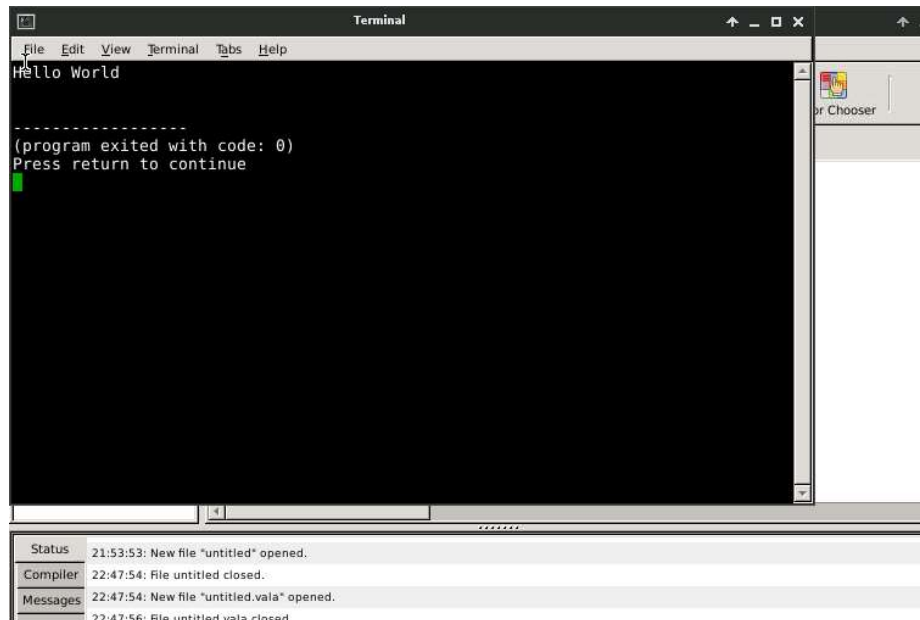
## การเขียนโปรแกรมพื้นฐาน 101

เอาล่ะเมื่อทำทุกอย่างเรียบร้อยแล้ว เราจะมาลองรันโปรแกรมของเรากัน  
โดยวิธีรันโปรแกรมนั้น มีได้ 3 ทางคือ

1. ใช้ทางลัด โดยกด F5
2. เข้าไปที่เมนู Build -> Execute (F5)
3. กดปุ่ม Execute ที่เป็นสัญลักษณ์เฟือง บนเมนู



แล้วก็จะได้พบกับ “Hello World” อย่างที่เราต้องการแล้ว



### แนวคิดในการพัฒนาโปรแกรม

แนวคิดในการพัฒนาโปรแกรมนั้น มีการพัฒนามายาวนาน โดยอาศัยพื้นฐานจากการพัฒนากลไกของเครื่องจักรกล ที่มีการทำงานที่ละขั้นตอนต่อกันไป

ซึ่งในสมัยแรกของระบบคอมพิวเตอร์นั้น(ช่วงก่อนปี 1936 ที่อลัน ทัวริงนำเสนอแนวคิดเรื่อง คอมพิวเตอร์สมัยใหม่) มันไม่มีความสามารถในการคำนวณมากนัก และยังไม่มีความคิดในการจัดเก็บข้อมูลอย่างเป็นรูปแบบอีกด้วย ทำให้งานหลักของมันยังคงเป็นเพียงทำงานตามขั้นตอนที่ถูกกำหนดไว้แล้ว โดยอาศัยการออกแบบ และพัฒนาโปรแกรมด้วยแนวคิด สเตต-แมคชีน (State-Machine) ที่มีการพึ่งพาการคำนวณน้อย และไม่มีการจัดการหน่วยความจำ

แต่หลังจากปี 1949 ซึ่งเป็นปีที่ แนวคิดคอมพิวเตอร์สมัยใหม่ ถูกทำให้กลายเป็นเครื่องคอมพิวเตอร์จริงๆ ที่ชื่อ EDSAC โดย EDSAC นั้นมีโครงสร้างคล้ายคอมพิวเตอร์สมัยนี้มาก มีทั้งหน่วยประมวลผล และหน่วยความจำในเวลาเดียวกันเอง ที่แนวคิดการพัฒนาโปรแกรมโดยใช้โฟลว์ชาร์ต ถูกนำมาใช้อย่างจริงจัง เนื่องจากคอมพิวเตอร์ EDSAC ถูกพัฒนามาบรรทัดประสงค์ให้คอมพิวเตอร์สามารถทำงาน และคำนวณอย่างซับซ้อนได้ ดังนั้นแล้วการออกแบบโปรแกรมด้วยโฟลว์ชาร์ตนั้นจึงเหมาะสมกับคอมพิวเตอร์สมัยใหม่มากกว่า

### การพัฒนาโปรแกรมแบบสเตต-แมคชีน(State-Machine)

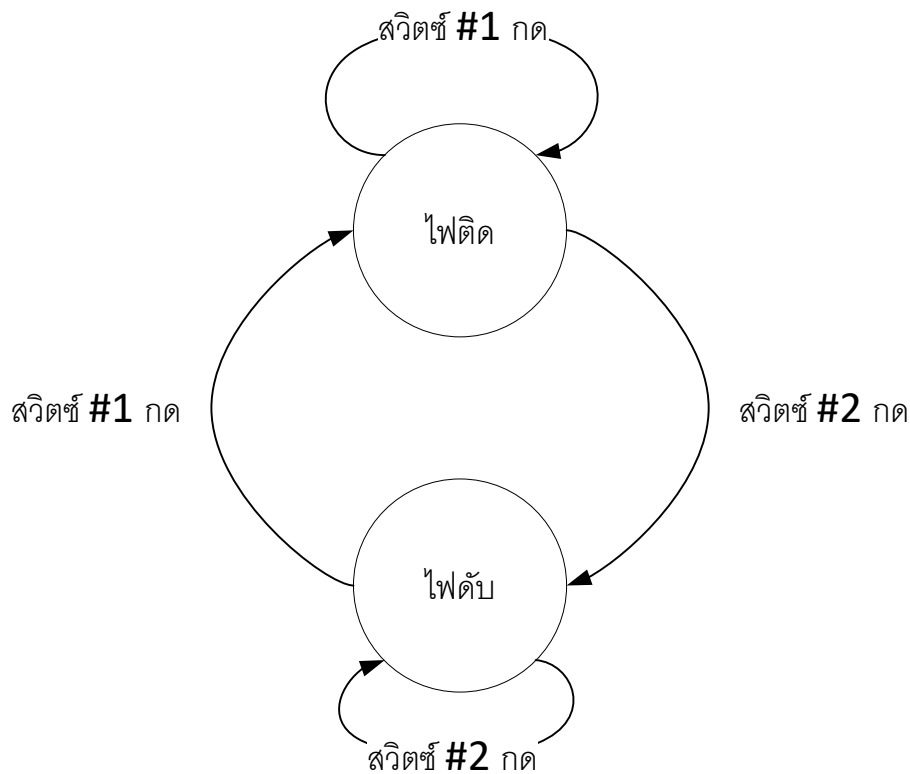
สเตต-แมคชีนนั้น ไม่มีสัญลักษณ์ และอะไรที่ซับซ้อน แนวความคิดของมันคือ การทำงานตามสถานะต่างๆของโปรแกรม ดังนั้นโครงสร้างโปรแกรมจึงประกอบไปด้วยเพียง 2 สัญลักษณ์หลัก คือ “การทำงาน” (Action) และ “สถานะ” (State) เช่น

เราต้องการพัฒนาโปรแกรม ให้หลอดไฟ LED ติดดับ จากการควบคุมด้วยสวิตช์ 2 ตัว โดยให้ตัวหนึ่ง กดเพื่อให้ไฟติด และอีกหนึ่งตัว กด เพื่อให้ไฟดับ

เราจะพบว่า ระบบของเรามีการ”ทำงาน” 2 อย่างคือ “ไฟติด” และ”ไฟดับ” ซึ่งถูกควบคุมจากสถานะของการ ”กด” ของสวิตช์ 2 ตัว



## การเขียนโปรแกรมพื้นฐาน 101



ตัวอย่างโค้ดจากสเตต-แมชชีนนี้

```
def LED_ON 1 : #Action เปิด LED
def LED_OFF 0: #Action ปิด LED
def SW_PSHD 1: #State เมื่อกดสวิตช์
def SW_RLSD 0: #State เมื่อปล่อยสวิตช์
def SW1_PIN 8: #Pin สำหรับสวิตช์#1
def SW2_PIN 9: #Pin สำหรับสวิตช์#2
def LED_PIN 7: #Pin สำหรับ LED

import wiringpi

sw1 = 0 #ตัวแปร sw1 สำหรับ สวิตช์#1
sw2 = 0 #ตัวแปรsw2 สำหรับสวิตช์#2
led = 0 #ตัวแปรled สำหรับ LED

def main():
    wiringpi.wiringPiSetup()

    print "Simplify State-Machine"
    print ""Switch#1 had pressed for turn LED ON
        Switch#2 had pressed for turn LED OFF""
    print "Pin 7 to LED"
    wiringpi.pinMode(LED_PIN,1)
    print "Pin 8 to Swich#1"
    wiringpi.pinMode(SW1_PIN,0)
    print "Pin 9 to Swich#2"
```

## การเขียนโปรแกรมพื้นฐาน 101

```
wiringpi.pinMode(SW2_PIN, 0)
while 1 :
    sw1 = wiringpi.digitalRead(SW1_PIN)    #อับเขตสถานะของสวิตช์#1 ไปเก็บไว้ที่ ตัวแปร sw1
    sw2 = wiringpi.digitalRead(SW2_PIN)    #อับเขตสถานะของสวิตช์#2 ไปเก็บไว้ที่ ตัวแปร sw2
    if sw1 == SW_PSHD :                    #ถ้า sw1 มีสถานะเท่ากับ SW_PSHD
        led = LED_ON                      #ไฟ LED ติด
    if sw2 == SW_PSHD :                    #ถ้า sw2 มีสถานะเท่ากับ SW_PSHD
        led = LED_OFF                     #ไฟ LED ดับ
    wiringpi.digitalWrite(LED_PIN, led)    #ให้ หลอดLED ทำงานตามสถานะของ led
return 0
```

### การพัฒนาโปรแกรมด้วยโฟลว์ชาร์ต

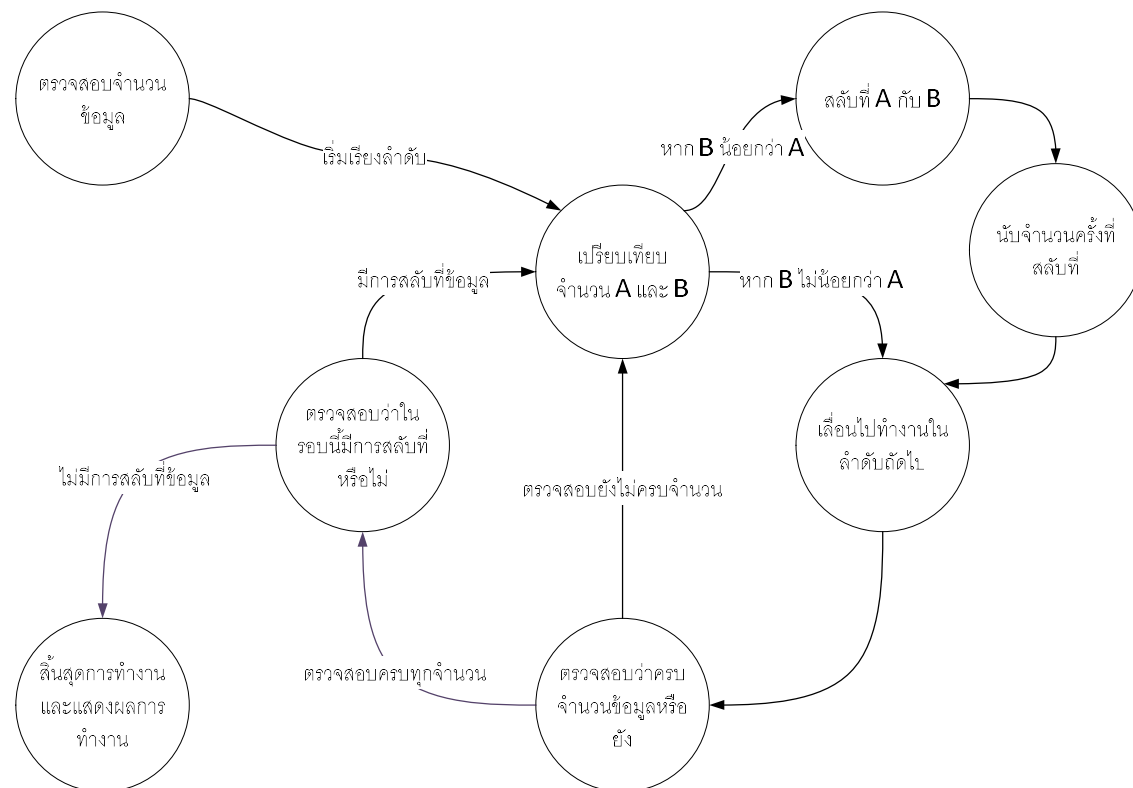
แนวความคิดในการใช้โฟลว์ชาร์ตในการพัฒนาโปรแกรมนั้น เกิดจากความต้องการใช้คอมพิวเตอร์เข้ามาช่วยในการทำงานวิจัย และศึกษา ในกระบวนการทางวิทยาศาสตร์ และวิศวกรรม เนื่องจากการพัฒนาโปรแกรมด้วยแนวคิดดั้งเดิมอย่าง สเตต-แมคชีนนั้น ไม่สามารถตอบสนองการพัฒนาโปรแกรมที่จำเป็นต้องมีการคำนวณทางคณิตศาสตร์ หรือมีกระบวนการสังเคราะห์ข้อมูลทางวิทยาศาสตร์ได้ เพราะความยากลำบากในการตรวจสอบกระบวนการทำงาน ซึ่งส่งผลโดยตรงต่อความน่าเชื่อถือของกระบวนการ เนื่องจากแนวคิดของสเตต-แมคชีนนั้น มีรูปแบบใช้งานอย่างจำกัด โดยมีเพียงแค่ สัญลักษณ์วงกลมที่สื่อแทนการทำงาน และเส้นลากที่สื่อแทนสถานะ ให้ใช้งาน ต่างจากแนวคิดของโฟลว์ชาร์ต ที่มีทางเลือกมากกว่า สัญลักษณ์ที่มากกว่า และมีความหมายที่ชัดเจนในแต่ละสัญลักษณ์ ทำให้สามารถทำความเข้าใจได้ง่ายกว่า เมื่อนำมาออกแบบโปรแกรมที่ซับซ้อน

การปรับเปลี่ยนโครงสร้างของคอมพิวเตอร์ และแนวคิดในการพัฒนาโปรแกรมนี้นำส่งผลอย่างมากต่อทิศทางการพัฒนาโปรแกรม เพราะทำให้เกิดโปรแกรมที่เรียกว่า อัลกอริทึม (Algorithm) ซึ่งเป็นชุดโปรแกรมที่ถูกออกแบบมาเพื่อให้ทำงานที่ซับซ้อนได้เอง เช่น การเรียงลำดับชุดตัวเลขจากน้อยไปมาก เป็นต้น แม้ว่าการใช้แนวคิดแบบ สเตต-แมคชีน จะสามารถทำงานได้เช่นเดียวกัน แต่การออกแบบนั้นกลับซับซ้อนมากเกินไปจนเกิดความจำเอน และยังตรวจสอบได้ยากอีกด้วย

## การเขียนโปรแกรมพื้นฐาน 101

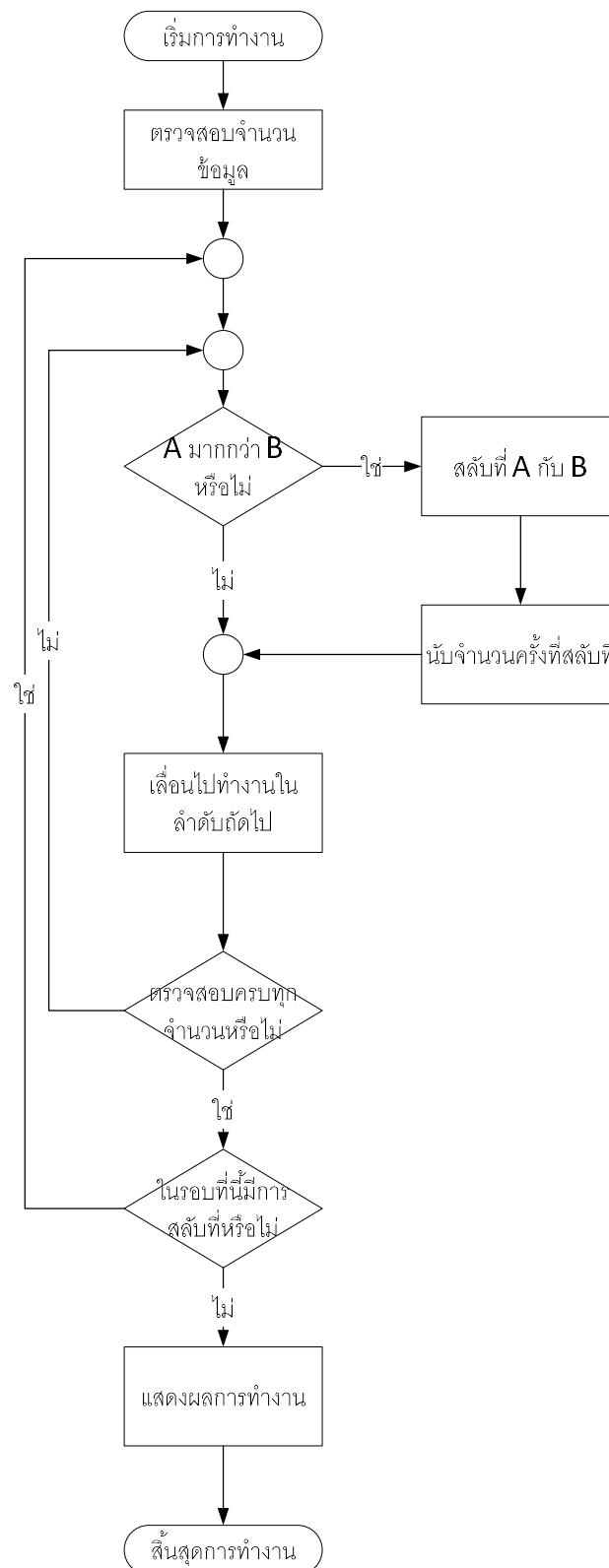
ตัวอย่างการออกแบบโปรแกรมเรียงลำดับตัวเลขจากน้อยไปมาก

สเตต-แมคชีน ในการเรียงลำดับตัวเลขจากน้อยไปมาก



## การเขียนโปรแกรมพื้นฐาน 101

โฟลว์ชาร์ตในการเรียงลำดับตัวเลขจากน้อยไปมาก



## การเขียนโปรแกรมพื้นฐาน 101

โค้ดตัวอย่างการเรียงลำดับเลขจำนวนจากน้อยไปมาก

```
numberTest = [5,2,3,4,2,7]
l = len(numberTest) #ตรวจสอบจำนวนชุดข้อมูล
s = 1 #สร้างตัวแปรไว้เก็บจำนวนการสลับที่

while s != 0 : #ทำงานจนกว่าจะไม่มีการสลับที่
    s = 0 #รีเซ็ตนับการสลับที่ใหม่
    for c in range (0 , l-1): #วนตรวจสอบข้อมูลจนชุดสุดท้าย
        a = numberTest[c] #เก็บค่าชุดปัจจุบันไว้ที่ a
        b = numberTest[c+1] #เก็บค่าชุดถัดไว้ที่ b
        if a > b : #เปรียบเทียบค่า A กับ B
            #สลับค่า A กับ B ถ้า A มากกว่า B
            numberTest[c] = b
            numberTest[c+1] = a
            s = s + 1 #นับจำนวนการสลับที่
        else :
            continue #ถ้า A ไม่มากกว่า B ให้ทำในข้อมูลชุด
            #ถัดไป
    else : #ถ้าไม่มีสลับที่แล้ว
        print "Mission Completed"
        print numberTest #แสดงข้อมูลที่มีการเรียงลำดับแล้ว
```