

# ม่านน้ำตกดิจิทัล

สุวิวัฒน์ วงษ์คำ<sup>1</sup>, นัฐนนท์ พัดโพธิ์<sup>1</sup>, ธีรยุทธ โหระนันท์<sup>2</sup>, อลงกต กองมณี<sup>1</sup>, ดีพร้อม สมเกียรติเจริญ<sup>2</sup>

<sup>1</sup>ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้

<sup>2</sup>คณะคณะเทคโนโลยีสารสนเทศและการสื่อสาร สถาบันเทคโนโลยีนานาชาติสิรินธร มหาวิทยาลัยธรรมศาสตร์  
ศูนย์บางกะดี

Email: sutivat\_nes@hotmail.com, nattanon.p338@gmail.com, teerayut@siit.tu.ac.th,  
alongkot@mju.ac.th, groovee123456@gmail.com

## บทคัดย่อ

รูปภาพหน้าจอบนม่านน้ำตกดิจิทัลจะแสดงผลด้วยรูปภาพหรือข้อความที่ป้อนเข้าไปจากหน้าเว็บ งานวิจัยนี้มีวัตถุประสงค์เพื่อหาอัตราส่วนความสูงของภาพที่เหมาะสมที่จะแสดงบนม่านน้ำตกดิจิทัล ซึ่งใช้ไมโครโปรเซสเซอร์ที่มีราคาต่ำมาควบคุมโซลินอยด์วาล์ว 128 ตัวเพื่อสร้างม่านน้ำตกดิจิทัลขนาด 2x2.6 ตารางเมตร และโซลินอยด์วาล์วจะควบคุมการเปิดปิดของเส้นหยดน้ำ เราใช้เวลา 4 เดือนในการหาอัตราส่วนของภาพที่เหมาะสมในการแสดงผลโดยความสูงของภาพจำเป็นต้องลดขนาดความสูงของภาพให้เหลือความสูงเพียง 1 ใน 6 ส่วน เพราะหลังจากหยดน้ำตกจากโซลินอยด์วาล์วรูปร่างของภาพจะเปลี่ยนไปตามแรงโน้มถ่วงของโลก ถ้าอัตราส่วนความสูงของภาพมากกว่านี้ยิ่งทำให้ภาพบนม่านน้ำตกยิดในทางกลับกันถ้าอัตราส่วนน้อยกว่านี้จะทำให้ภาพที่แสดงบนม่านน้ำตกดิจิทัลบิดเบี้ยว

## ABSTRACT

Displayed images on graphic waterfalls are effected by the input settings of the image or texts from web page. This research aims to find the optimal ratio of the height between the input image and displayed image. A low cost microprocessor was used to control 128 solenoid valves to generate 2 by 2.6 square meter screen. Each solenoid valve switches one of 128 vertical water lines on-and-off. We conduct the experiment for 4 months with different image size ration. The findings show that the height of the input image must be resized to one sixth of image to be display. This is because when the water drops fall, its shapes became more taper from the gravity force. Higher ration will stretch the displayed image whereas smaller ration will shrink the displayed images.

**คำสำคัญ**-- ม่านน้ำตกดิจิทัล; น้ำตกกราฟฟิก; ม่านกราฟฟิก; น้ำตกดิจิทัล; รัสเบอร์รี่พาย

## 1. บทนำ

น้ำตกและม่านน้ำเป็นรูปแบบทางเลือกหนึ่งที่สามารถนำมาใช้เพื่อความบันเทิงหรืองานอีเว้นท์ต่างๆ เราสามารถใช้ม่านน้ำตกดิจิทัลเพื่อการตกแต่งเล็ก ๆ น้อย ๆ ภายในบ้านและอาคารหรือเพื่อการแสดงกลางแจ้งในเหตุการณ์ใหญ่ๆ ได้ สิ่งที่น่าสนใจในงานม่านน้ำตกดิจิทัลคือสามารถแสดงข้อความ ตัวเลข ภาพ และรูปแบบต่างๆ ม่านน้ำตกดิจิทัลนี้เหมาะกับผู้คนที่มียุคที่ทันสมัย ซึ่งขึ้นอยู่กับวัฒนธรรมของแต่ละสถานที่ มันมีผลต่อการสร้างพื้นที่สภาพแวดล้อมที่ให้ความสวยงามและความสงบทางจิตใจ

แรงบันดาลใจของเรามาจากน้ำตกกราฟฟิกดิจิทัลใน สถานีรถไฟใต้ดินโอซาก้าที่ประเทศญี่ปุ่น เป็นน้ำตกกราฟฟิกดิจิทัลที่จะทำหน้าที่เป็นนาฬิกาแสดงเวลาขณะนั้นๆ และแสดงหลากหลายของรูปแบบการประดับและข้อความระบุชื่อของสถานี มันมีความแม่นยำในเรื่องของเวลาเช่นเดียวกับรายละเอียดของภาพ

ม่านน้ำตกดิจิทัลทำงานโดย Raspberry pi เป็นตัวสั่งการและควบคุมโซลินอยด์วาล์วกับท่อน้ำให้เปิดปิดตามคำสั่งตั้งแต่ต้นจนจบ ควรจะเตรียมน้ำให้เพียงพอสำหรับกระบวนการน้ำที่ใช้ในการหมุนเวียนระบบ โซลินอยด์วาล์วทำงานในรูปแบบการเปิด / ปิดที่เชื่อมโยงไปยังข้อมูลไบนารีที่ได้ทำการแปลงข้อมูลที่รับคำสั่งเข้ามา เราสามารถที่จะใส่ข้อมูลประเภทภาพ ตัวอักษร ที่หลากหลายต่างๆและแสดงผลโดยการหยดน้ำได้ แต่ก็มาพร้อม กับข้อจำกัด เกี่ยวกับขนาดของม่านน้ำหลายประเด็นที่จะต้องตระหนักถึงคือ

- 1) ขนาดของม่านน้ำตกดิจิทัล
- 2) จำนวนของโซลินอยด์วาล์ว
- 3) ช่องว่างระหว่างของแต่ละโซลินอยด์วาล์ว
- 4) ประเภทของโซลินอยด์
- 5) การไหลของน้ำและการควบคุมน้ำในระบบ
- 6) เครื่องสูบน้ำ

แต่ละประเด็นที่สำคัญเหล่านี้จะส่งผลกระทบต่อการทำงานของระบบอย่างมากและจะได้รับการวิเคราะห์ในเนื้อหาถัดไป

## 2. งานวิจัยที่เกี่ยวข้อง

ในขั้นตอนวิธีการหา Threshold มีวิธีการวิจัยที่คล้ายกัน วิธีการนี้ได้ใช้กันอย่างแพร่หลายเนื่องจากการคำนวณที่มั่นคงและเรียบง่ายวิธีการจะทำงานด้วยค่าสีเทาของภาพเท่านั้น ขั้นตอนการ Threshold จะพิจารณาข้อมูลสีเทาระดับพิกเซลกำหนดเกณฑ์ขั้นต่ำเป็นวิธีงานสุดในการแบ่งส่วนภาพจากรูปภาพสีเทา Threshold สามารถนำมาใช้ในการสร้างภาพไบนารี(Shapiro, et al. 2001:83) วิธี Threshold ที่มีระดับความเข้มอยู่ระหว่างกลุ่มทั้งสองของ Histogram ซึ่งค่า Threshold ที่ได้จะอยู่ระหว่าง 0-255 เท่านั้น Threshold จะถูกนำไปเพื่อเปรียบเทียบค่าของแต่ละพิกเซล หากค่า  $f(x, y)$  น้อยกว่า Threshold จุดพิกเซลนั้นจะถูกปรับให้เป็นสีดำหรือส่วนของวัตถุ และหากค่า  $f(x, y)$  มากกว่าหรือเท่ากับ Threshold จุดพิกเซลนั้นจะถูกปรับให้เป็นสีขาวหรือส่วนของพื้นหลัง สามารถเขียนแทนด้วยสมการดังนี้

$$f_{thr}(x, y) = \begin{cases} 0 & f(x, y) < Threshold \\ 1 & f(x, y) \geq Threshold \end{cases} \quad (1)$$

โดยกำหนดให้

0 คือ สีขาว ซึ่งเป็นส่วนของพื้นหลัง

1 คือ สีดำ ซึ่งเป็นส่วนของวัตถุ

ในทฤษฎี Image Denoising ประกอบด้วยเทคนิคการเบลอภาพที่เรียกว่า Gaussian Blurring, Median Blurring โดยจะใช้วิธีการลบวัตถุ รูปภาพ หรือสิ่งรบกวนในภาพที่ไม่ต้องการ ซึ่งเราจะนำเอาพิกเซลที่อยู่รอบๆผ่านกระบวนการที่เรียกว่า Gaussian Weighted ซึ่งทำให้ขนาดรูปภาพเล็กลง ค่ามัธยฐานของแต่ละอันจะถูกแทนที่ตรงกลางจตุรบกวนและสิ่งรบกวนจะถูกลบในตำแหน่งที่เหมาะสม

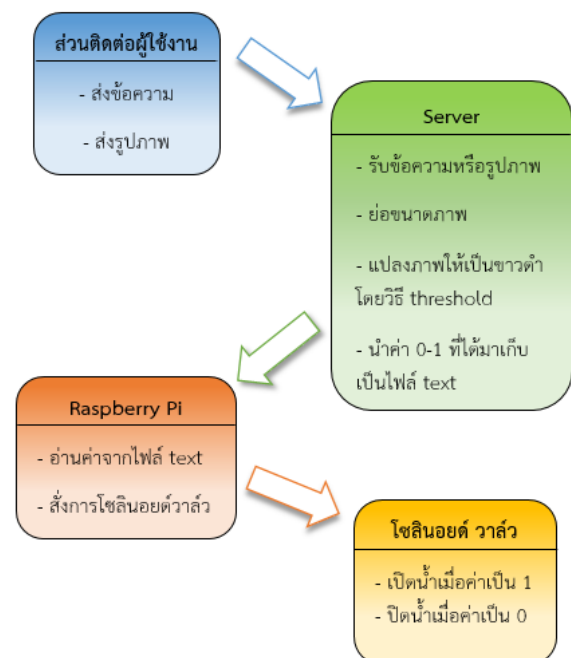
Graphic Waterfall ควบคุมโดย Raspberry Pi มีความกว้าง 1.5 เมตรความสูง 2.5 เมตร ใช้ โซลินอยด์วาล์ว 32 ตัวเพื่อแสดง

หน้าจอขนาด 32 พิกเซล มันสามารถแสดงรูปแบบและข้อความแล้วส่งไปแสดงที่ม่านน้ำตกในการออกแบบ จะมีส่วนที่พัฒนามากขึ้นดังนี้

- เพิ่มขนาดของ โซลินอยด์วาล์วจากที่มีขนาด 32 พิกเซล เป็น 128 พิกเซล
- เพิ่มฟังก์ชันการหารูปภาพแล้วนำภาพที่ได้ไปแปลงเพื่อแสดงผลบนม่านน้ำตก
- เพิ่มฟังก์ชันการวาดรูปหรือข้อความ เพื่อนำมาแสดงผลบนม่านน้ำตก

## 3. ระบบและการออกแบบ

Flask เป็นเว็บเฟรมเวิร์คขนาดเล็กที่เปิดให้ใช้งานได้ไม่มีข้อจำกัดใช้ภาษา python ในการพัฒนาเพื่อที่จะประมวลผลและทำงานบน Raspberry pi ได้ทั้งหมดโดยไม่ต้องอาศัยเทคโนโลยีอื่นๆเข้ามาช่วย เมื่อผู้ใช้ทำการพิมพ์ตัวอักษรหรือเลือกภาพ โปรแกรมจะเริ่มทำงานโดยการตรวจสอบแยกประเภทว่าเป็นตัวอักษรหรือรูปภาพ หากได้เป็นตัวอักษรจะทำการแปลงเป็นภาพโดยเทคโนโลยี Canvas และทำการแปลงรูปภาพให้เป็นไฟล์รูปแบบ text ที่มีค่า 0 กับ 1 แต่ถ้าตรวจสอบแล้วได้เป็นรูปภาพจะเข้าสู่กระบวนการแปลงรูปภาพเป็นไฟล์ text ที่มีค่า 0 กับ 1 ซึ่งมีส่วนภาพการทำงานทั้งหมดของระบบดังรูปภาพที่ 1



รูปที่ 1. แผนภาพแสดงขั้นตอนการทำงานของระบบ

การแปลงรูปภาพจะมีขั้นตอนคือการปรับขนาดของภาพให้มีความกว้างขนาดเท่ากับโครงสร้างคือ 128 บิตและความสูงปรับตามความเหมาะสมของภาพ ทำการกลับภาพต่อด้วยการแปลงภาพให้เป็นสีเทาและแปลงเป็นสีขาวดำ จะได้ค่าที่เป็น 0-255 เป็นสีขาว – ดำ ดังรูปภาพที่ 2



รูปที่ 2. รูปภาพที่ทำการแปลงเป็นขาว-ดำแล้ว

และทำการแปลงค่า 0-255 ให้เป็นค่า 0 กับ 1 หลังจากนั้นทำเป็นอาร์เรย์เขียนลงไฟล์ text ม่านน้ำตกดิจิทัลเป็นการแสดงผลภาพดิจิทัล จึงจำเป็นต้องแปลงภาพให้เป็นไบนารีเพื่อที่จะสามารถทำการตั้งค่าการแสดงผลได้ คือโดยการทำให้ค่า 0 กับ 1 ที่แปลงมาได้นั้นเป็นตัวกำหนดการทำงานของโซลินอยด์วาล์วในการเปิด-ปิดน้ำ

ในส่วนนี้จะมีเทคนิคของการแปลงภาพที่มีค่าน้อยหรือสัญญาณรบกวนที่เกิดขึ้นในภาพ ทำให้สีภายในภาพเกิดความไม่สม่ำเสมอ ทำให้ภาพที่ได้มีจุดๆ ปรากฏขึ้นในภาพ เกิดจากหลายสาเหตุ การใส่ค่าส่งรูปภาพเข้ามานั้น รูปภาพจะมีค่าน้อยซึ่งที่มาจากรูปภาพจะส่งผลต่อการแปลงรูป จึงจะต้องนำรูปภาพนั้นมาทำการลดน้อยซ์เพื่อการแปลงรูปไบนารี ดังรูปภาพที่ 3



รูปที่ 3. การลดสัญญาณรบกวนของรูปภาพ

ในการแปลงรูปภาพจะมีส่วนการลดขนาดของภาพ ซึ่งมีสองขั้นตอนคือ

- ลดขนาดรูปภาพให้มีความกว้าง 128 พิกเซล และความสูง ตามสัดส่วนของภาพ

- ย่อความสูงของภาพให้เหลือเพียง 1 ใน 6 ส่วนของส่วนสูงปกติภาพ

ในการปล่อยน้ำที่มีแรงดันน้อย จะมีอัตราแรงต่ำจากหัวโซลินอยด์วาล์วที่ปล่อย พอผ่านไประยะหนึ่ง หยดน้ำจะร่วงลงด้วยอัตราเร็วเท่าคืบแรงโน้มถ่วง ซึ่งระยะที่อัตราแรงคืบนั้นจะอยู่ที่ไม่เกิน 20 เซนติเมตรหลังจากถูกปล่อยจากหัวโซลินอยด์วาล์ว ซึ่งอัตราเร็วการหล่นของหยดน้ำมีผลต่อรูปร่างของรูปภาพที่แสดงบนม่านน้ำตกดิจิทัล เมื่อความเร็วในการหล่นของหยดน้ำคงที่แล้ว รูปภาพที่แสดงผลออกมาก็จะคงที่ด้วย

โซลินอยด์วาล์วที่ระบบใช้มีความเร็วในการหยดน้ำประมาณ 40 มิลลิเมตรต่อวินาที ที่จะทำให้เห็นเป็นหยดน้ำ โดยขึ้นอยู่กับโซลินอยด์วาล์วแต่ละรุ่นที่ใช้ ซึ่งหมายความว่าความเร็วที่เร็วที่สุดในการเปิดปิดโซลินอยด์วาล์วของเราสามารถทำงานได้จะอยู่ที่ประมาณ 40 มิลลิเมตรต่อวินาที ซึ่งเราได้ใช้โซลินอยด์วาล์วดังรูปภาพที่ 4

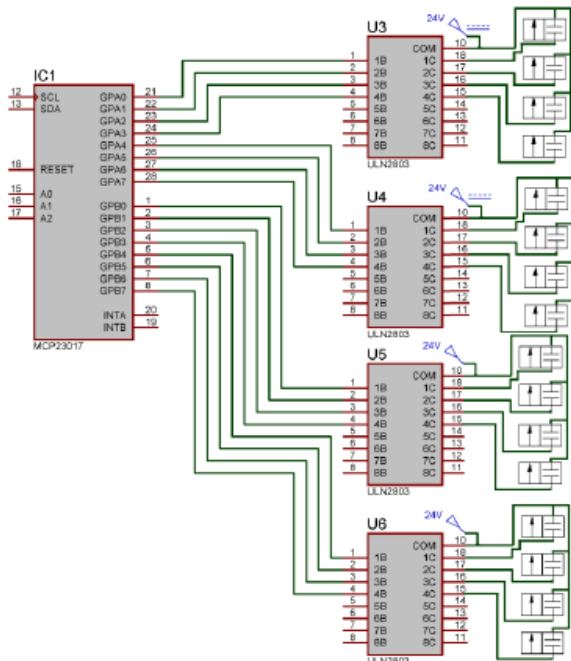


รูปที่ 4. โซลินอยด์วาล์วที่ใช้

หลังจากจัดตำแหน่งโซลินอยด์วาล์วทั้ง 128 วาล์วให้อยู่ในตำแหน่งที่มีระยะห่างระหว่างวาล์วอยู่ที่ 1.4 เซนติเมตรตามความเหมาะสมในด้านบนโดยมีความกว้างของฐานวางโซลินอยด์วาล์วโดยจะอยู่ที่ 2 เมตรและสูงจากพื้น 2.6 เมตร เมื่อปล่อยน้ำตกลงมาถึงแอ่งน้ำจะมีความสูงเหลืออยู่ที่ 2.3 เมตรที่ใช้ในการแสดงม่านน้ำตก ส่วนฐานในการวางโซลินอยด์วาล์วได้ใช้สมาร์ทบอร์ดยึดติดไว้ด้วยน็อตและเจาะรูเพื่อวางท่อปล่อยน้ำออกในส่วนกลางของโครงสร้างแล้วนำฐานวางบนโครงเหล็กยึดติดกับผนังไว้

ในส่วนของวงจรไฟฟ้านั้น จะใช้ i/o pi 32 channel port expander เพื่อใช้ขยายการคุมของ raspberry pi และใช้ IC

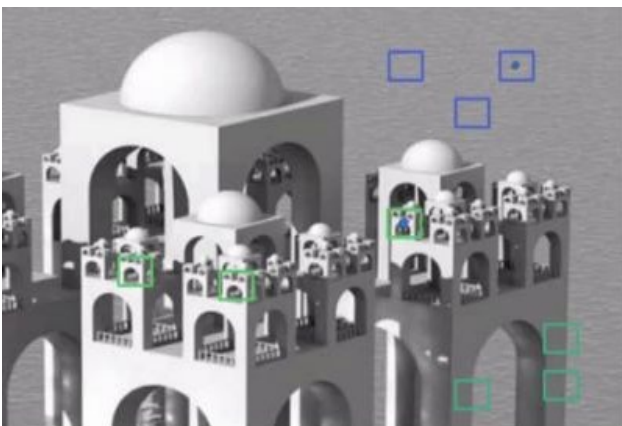
ULN 2803 เพื่อคุมโซลินอยด์วาล์วอีกที โดยจะมีผังวงจรตามรูปภาพที่ 5



รูปที่ 5. ผังวงจรไฟฟ้าที่ใช้ควบคุมโซลินอยด์วาล์ว 16 ตัว

#### 4. ผลลัพธ์ของระบบ

ในส่วนติดต่อผู้ใช้งาน จะสามารถใส่ข้อความ เลือกรูปภาพและวาดภาพได้ ซึ่งจะเป็น เว็บเบส โดยใช้ html และ bootstrap ในการพัฒนา ประมวลผลโดยใช้ Flask ในการส่งข้อความหรือรูปภาพ หากเป็นรูปภาพจะทำการลด noise ของภาพซึ่งได้แนวคิดจาก Image denoising จะทำการเฉลี่ยรูปภาพเพื่อลดกระแสบกวนของภาพ ซึ่งเป็นจุดเล็กๆ อาจจะทำให้ภาพที่ได้ออกมามีความผิดเพี้ยน ซึ่งจะเห็นได้จากรูปภาพตัวอย่างที่ 6



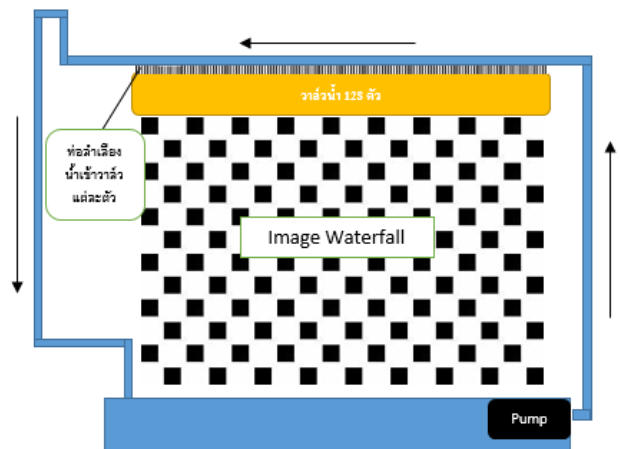
รูปที่ 6. ภาพตัวอย่างการเฉลี่ยรูปภาพเพื่อลดกระแสบกวนของภาพ

กรอบสีฟ้าในภาพที่มีลักษณะคล้ายกันและกรอบสีเขียวมีลักษณะคล้ายกันดังนั้นเราจึงใช้พิกเซลใช้หน้าต่างเล็กๆ รอบๆ ค้นหาสำหรับกรอบที่คล้ายกันในภาพเฉลี่ยหน้าต่างทั้งหมดและแทนที่พิกเซล มันต้องใช้เวลามากขึ้นเมื่อเทียบกับเทคนิคการเบลอภาพ แต่ผลของมันเป็นที่ดีมาก ก่อนจะทำการแปลงภาพให้เป็นภาพขาว-ดำ

การจะแปลงภาพให้เป็นภาพขาว-ดำ โดยใช้หลักการของ Threshold ซึ่งจะได้ค่าตัวเลขของภาพเป็นเลข 0 ถึง 255 ของแต่ละพิกเซลภาพ จากนั้นจะทำการแปลงชุดตัวเลข 0 ถึง 255 ให้กลายเป็นชุดตัวเลข 0 – 1 เพื่อที่จะสั่งการเปิด-ปิดวาล์ว

การทำงานของวาล์วจะแสดงผลที่ละแถว ความกว้างของภาพจะกว้าง 128 พิกเซล ชุดของค่า 0-1 ก็จะเท่ากับ128ตัวเลขเช่นกันของแต่ละชุด ในการอ่านภาพจะอ่านจาก ล่างขึ้นบน เพื่อจะได้ลำดับภาพออกมาตามแบบรูปจริง

ระบบจะมีการวนน้ำ เพื่อนำน้ำกลับมาใช้ใหม่อีกครั้ง เพื่อลดความสิ้นเปลืองของน้ำ ซึ่งน้ำทั้งหมดที่ใช้จะเป็นอ่างน้ำ โดยจะมีปั้มน้ำเพื่อไปเลี้ยงระบบ ดังรูปภาพที่ 7



รูปที่ 7. โมเดลโครงสร้างการทำงานของระบบน้ำ

จากรูปจะเห็นท่อน้ำด้านบนซ้ายมีการหักงอขึ้นไปเพื่อเพิ่มแรงดันของน้ำทำให้แรงดันน้ำที่เข้าวาล์วแต่ละตัวเท่ากัน จึงไม่ทำให้วาล์วตัวใดตัวหนึ่งแรงหรือเบากว่าตัวอื่นๆ

ทฤษฎีต่างๆที่กล่าวมานั้น สามารถนำไปใช้ได้จริง ซึ่งได้ทำการทดสอบและทำออกมาเรียบร้อยแล้ว สามารถปรับใช้ได้ตามความเหมาะสม ตามรูปภาพที่ 8



รูปที่ 8. ภาพตัวอย่างโมเดลที่สำเร็จแล้ว

### เอกสารอ้างอิง

- [1] Wilhelm Burger, and Mark Burge, Digital image processing in Java, 2nd ed. London : Springer, 2007.
- [2] รศ.ดร.สมเกียรติ อุดมหาระยากุล. การประมวลผลภาพดิจิทัลเบื้องต้น(Fundamentals of Digital Image Processing). พิมพ์ครั้งที่ 1. กรุงเทพฯ : สำนักพิมพ์ท็อป, 2554.
- [3] Bhinyaporn Pattanasethanon, Deeprom Somkiadcharoen, and Kamonnate Cheevaprawatdomrong. "Graphic Waterfall Controlled by Raspberry Pi". (ธันวาคม 2556). หน้า 1-26.
- [4] Nattida Leesom and Olarik Surinta. "Thai Handwritten Character Segmentation from Digital Image Documents." [ระบบออนไลน์]. แหล่งที่มา [http://www.wbi.msu.ac.th/file/595/doc\\_6\\_paper.doc](http://www.wbi.msu.ac.th/file/595/doc_6_paper.doc)
- [5] Doxygen. "Image Denoising." [ระบบออนไลน์]. แหล่งที่มา [http://docs.opencv.org/trunk/d5/d69/tutorial\\_py\\_non\\_local\\_means.html](http://docs.opencv.org/trunk/d5/d69/tutorial_py_non_local_means.html).
- [6] Shapiro. "Thresholding (image processing)." [ระบบออนไลน์]. แหล่งที่มา [https://en.wikipedia.org/wiki/Thresholding\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Thresholding_(image_processing)).
- [7] Armin Ronacher. "Flask web development." [ระบบออนไลน์]. แหล่งที่มา <http://flask.pocoo.org/>.
- [8] w3schools.com. "HTML canvas drawImage." [ระบบออนไลน์]. แหล่งที่มา [http://www.w3schools.com/tags/canvas\\_drawimage.asp](http://www.w3schools.com/tags/canvas_drawimage.asp)

- [9] Tim Hornyak. "Osaka Station fountain displays time, art in water." [ระบบออนไลน์]. แหล่งที่มา <https://www.cnet.com/news/osaka-station-fountain-displays-time-art-in-water/>
- [10] IBOOKENGINEERING. "การดึงวัตถุพื้นหน้าออกจากพื้นหลังโดยใช้ Threshold." [ระบบออนไลน์]. แหล่งที่มา <https://nextsoftwares.wordpress.com/2014/05/22/การดึงวัตถุพื้นหน้าออก/>
- [11] kunniiz. "ภาพ นอยซ์ คืออะไร." [ระบบออนไลน์]. แหล่งที่มา <http://www.taklong.com/beginner/show-beginner.php?No=484038>