

การพัฒนางาน IoT (Internet of Things) เพื่อใช้งานสมาร์ต โฟนหรือแท็บเล็ตควบคุมอุปกรณ์ เครื่องใช้ฟ้า แสดงค่าของตัวตรวจจับทาง ใกลผ่านเครือข่ายอินเทอร์เน็ต เป็นหนึ่งในการเรียนรู้ที่ผู้เรียน ระบบสมองกลฝังตัวสมัยใหม่ควรให้ความสำคัญ ในบทนี้นำเสนอถึงแนวทางและตัวอย่างการพัฒนา อุปกรณ์ IoT ที่ใช้ WIO Terminal เป็นส่วนประกอบหลักร่วมกับแอปพลิเคชั่น **Blynk** เพื่อช่วยให้เกิด การพัฒนาอุปกรณ์ IoT ได้อย่างรวดเร็ว และเหมาะสำหรับผู้เริ่มต้นที่ต้องการไปต่อยอดใช้งานจริง

11.1 รู้จักกับ IoT

Internet of Things คำนี้เกิดขึ้นมาตั้งแต่ปี ค.ศ. 1999 โดย Kevin Ashton แห่ง MIT's Media center เขาได้นำเสนอแนวคิดว่า IoT คือ การนำสิ่งของต่างๆ ไม่ว่าจะเป็นคอมพิวเตอร์, เครื่องจักร และ ตัวตรวจจับมาเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต เพื่อรายงานสถานะการทำงาน ข้อมูล และรับรู้คำสั่ง ควบคุม

IoT หรือ Internet of Things หมายถึง เทคโนโลยีที่ก่อให้เกิดการเชื่อมโยงกันของสิ่งของ ผู้คน ข้อมูล และการบริการเข้ากับเครือข่ายอินเทอร์เน็ต ปัจจัยสำคัญในการทำให้เกิด IoT ได้คือ การบรรจ อุปกรณ์สมองกลฝังตัวหรือ embedded system device เข้าไปใน "สิ่งของ" หรือเครื่องมือ เครื่องใช้ต่างๆ มีตัวตรวจจับหรือเซนเซอร์เพื่อตรวจวัดค่าที่สนใจ แล้วส่งมายังส่วนสมองกล เพื่อส่งต่อมายังส่วน ประมวลผลกลางและฐานข้อมูลผ่านเครือข่ายอินเทอร์เน็ต ในส่วนหลังนีมีชื่อเรียกด้วยศัพท์สมัยใหม่ ว่า **คลาวด์เซิรฟเวอร์ (cloud server)**

ด้วยการนำอุปกรณ์สมองกลฝังตัวบรรจุลงใน "สิ่งของ" ต่างๆ ทำให้ "สิ่งของ" เหล่านั้นทำงาน ในแบบอัจฉริยะได้ อุปกรณ์เกรื่องใช้ต่างๆ ในบ้าน ในโรงงาน ในที่ทำงาน ในยานหนะ ล้วนแล้วแต่ใช้ ระบบสมองกลฝังตัวมากขึ้น ทำให้มันทำงานได้ด้วยตัวเอง และ/หรือรวมเข้าเป็นส่วนหนึ่งของระบบใหญ่ เกิดการเชื่อมโยงการทำงานเป็นระบบได้



รูปที่ 11–1 ส่วนประกอบและการทำงานของ IoT เบื้องต้น

```
ส่วนประกอบของ IoT มีดังนี้
```

1. สิ่งของ

- 2. อุปกรณ์ (ตัวควบคุม, ตัวตรวจจับ และอุปกรณ์ขับโหลดหรืออุปกรณ์เอาต์พุต)
- 3. ระบบเชื่อมต่ออินเทอร์เน็ต (จะเป็นแบบมีสายหรือไร้สายก็ได้)
- 4. ข้อมูล
- 5. ระบบจัดการฐานข้อมูลคลาวด์เซิรฟเวอร์ (Cloud server)

การทำให้"สิ่งของ"ทำงานร่วมกันผ่านเครือข่ายอินเทอร์เน็ต จึงทำให้เกิคนิยามของเทคโนโลยีนี้ขึ้น Internet of Things หรือ IoT เป็นการขยายขอบเขตการทำงานของอินเทอร์เน็ตให้กว้างและลึกลงไปถึง การเชื่อมต่อเพื่อสื่อสารข้อมูลกับ "สิ่งของ" ทำให้เกิดการรับส่งข้อมูลและตอบสนองในแบบทุกที่ ทุกเวลา และทุกสิ่งของได้ในที่สุด

11.2 มารู้จักกับ Blynk2 หรือ Blynk IoT

ปัจจุบันนี้การพัฒนางาน IoT หรือ Internet of Things ที่มีการใช้งานสมาร์ตโฟนหรือแท็บเล็ต ควบคุมอุปกรณ์เครื่องใช้ฟ้า แสดงค่าของตัวตรวจจับทางใกลผ่านเครือข่ายอินเทอร์เน็ตทำได้ง่ายและ รวดเร็ว เนื่องจากมีตัวช่วยที่ดี เด่น และฟรี ในชื่อ Blynk App

Blynk อ่านว่า"บลิ้ง" คือ ชุดของแอปพลิเคชั่นที่ทำให้การสร้างงาน IoT ทำได้ง่ายอย่างเบ็คเสร็จ มีการเชื่อมต่อกับอุปกรณ์ที่อยู่ไกลผ่านเครือข่ายอินเทอร์เน็ต โดยใช้สมาร์ตโฟนเป็นอุปกรณ์หลัก ในการติดต่อกับผู้ใช้งานและอุปกรณ์ควบคุมปลายทาง ผู้พัฒนา IoT ไม่ต้องจัดเตรียมโปรแกรมอื่นๆ ไม่ว่าจะเป็นเว็บเซิร์ฟเวอร์, หน้าเว็บแสดงผลและควบคุม รวมถึงซอฟต์แวร์เพื่อการเชื่อมต่อใดๆ

Blynk เป็นผลงานของกลุ่มคนรุ่นใหม่ที่เสนอโครงการเข้าใน Kickstarter โดยผู้ก่อตั้งคือ **Mr.** Pavel Baiborodin โดยระคมทุนได้ 49,000 เหรียญสหรัฐ (US\$) Blynk ได้เปิดตัวใช้งานตั้งแต่เดือน พฤษภาคม ค.ศ. 2015



รูปที่ 1 1–2 แสดงส่วนประกอบต่าง ๆ Blynk ในเวอร์ซันใหม่ เพื่อช่วยให้การพัฒนางาน IoT เป็นไปได้ สำหรับ ทุกคนที่สนใจ โดยไม่จำกัดอยู่ที่สมาร์ตโฟนเท่านั้นอีกต่อไป ผู้ใช้งานสามารถติดต่อผ่านทางเว็บไซต์ โดยใช้อุปกรณ์ใดก็ได้ไม่ว่าจะเป็นสมาร์ตโฟน แท็บเล็ต หรือคอมพิวเตอร์



รูปที่ 11–3 ภาพแสดงความง่ายของการพัฒนางาน loT กับ Blynk loT App

Blynk ช่วยให้การพัฒนางาน IoT ง่าย โดยผู้พัฒนาไม่จำเป็นที่ต้องมีความรู้ด้านระบบฮาร์ดแวร์ และซอฟต์แวร์ทางกอมพิวเตอร์มากมาย ตัวแอปพลิเกชั่นใช้งานกับฮาร์ดแวร์ที่เป็นที่นิยม ทั้งบอร์ด Arduino, ESP8266, ESP32, Raspberry Pi, SAMD21 เป็นต้น

ปัจจุบัน Blynk ได้พัฒนามาเป็นเวอร์ชัน 2 ในตอนแรกเรียกว่า **Blynk2** ต่อมาทางผู้พัฒนาได้ ปรับ Blynk App ในเวอร์ชันแรกเป็น **Legacy Blynk** และหยุดการให้บริการในปีพ.ศ. 2565 ส่วน Blynk2 ได้ถูกเรียกชื่อใหม่ว่า **Blynk IoT** แต่ในเว็บไซต์ของผู้พัฒนาจะเรียกชื่อเพียงว่า **Blynk** ดังนั้น ในเอกสารนี้จะยึดชื่อของแพลตฟอร์มตามการอัปเดตล่าสุดของผู้พัฒนา

Blynk ในเวอร์ชันแรกได้รับความนิยมสูง เนื่องจากใช้งานง่าย เมื่อปรับปรุงเป็นเวอร์ชัน 2 มี การเปลี่ยนแปลงหลายอย่าง เช่น การสร้างและใช้งานแดชบอร์ด (dashboard) ผ่านเว็บไซต์ได้ การจำกัดจำนวนวิดเจ็ต (widget) ที่ใช้งานสำหรับรุ่นฟรีเป็น 30 ตัวแทนการให้ค่าพลังงาน และ รูปแบบการตั้งก่าที่แตกต่างไปจากเดิม กอปรกับการหยุดให้บริการ Blynk เซิร์ฟเวอร์รุ่นเดิมในปี พ.ศ. 2565 การเรียนรู้เพื่อใช้งานกับ Blynk App ในเวอร์ชันใหม่จึงเป็นสิ่งที่นักเล่น นักทดลอง นักพัฒนา รวมถึงนักเรียน นิสิต นักศึกษาที่ใช้งาน Blynk ในการพัฒนาอุปกรณ์ IoT ควรดำเนินกา

11.2.1 ที่สุดของความง่ายในการใช้งาน Blynk

Blynk มีขั้นตอนในการพัฒนาที่สุดแสนง่ายใน 3 ขั้นตอน คือ

(1) **หยิบแล้ววาง (Drag and Drop)** คือ การหยิบอุปกรณ์ที่ต้องการมาวางบนพื้นที่ทำงาน ซึ่งก็คือ หน้าจอแสดงผลของสมาร์ต โฟน (2) **ดาวน์โหลดไลบรารี (Download library)** คือ การดึงไฟล์สนับสนุนของ Blynk มาใส่ ในโปรแกรม สำหรับในโปรแกรม Arduino IDE จะมีไลบรารีสำหรับใช้งานกับ Blynk โดยเฉพาะซึ่ง ผู้พัฒนาโค้ดสามารถดาวน์โหลดไปติดตั้งและใช้งานได้ฟรี

(3) อัปโหลดโค้ด (Upload Sketch) คือ การส่งโค้คไปยังบอร์คไมโครคอนโทรลเลอร์

นั่นคือขั้นตอนหลักๆ ที่ง่ายๆ สำหรับกระบวนการพัฒนาอุปกรณ์ IoT โดยใช้อุปกรณ์ฮาร์ดแวร์ เป็นบอร์ดไมโครคอนโทรลเลอร์อย่างบอร์ด WIO Terminal ซึ่งใช้ไมโครคอนโทรลเลอร์ ATSAMD21 และพัฒนาโปรแกรมภาษา C/C++ ด้วย Arduino IDE ด้วยความสามารถของแอปพลิเคชั่น Blynk IoT ช่วยให้ผู้สนใจทุกคนสามารถพัฒนางาน IoT เองได้โดยไม่จำเป็นต้องมีทักษะทางการเขียนโปรแกรม มากมาย

11.2.2 การพัฒนางาน IoT โดยใช้ Blynk

ประกอบด้วยอุปกรณ์ทั้งหมด 3 ส่วนคือ

 Blynk.App เป็นแอปพลิเคชั่นและเว็บเซิร์ฟเวอร์ที่ติดตั้งบนสมาร์ตโฟนหรือแท็บเล็ต ทั้งบนระบบปฏิบัติการ iOS และ Android รวมถึงใช้งานผ่านเว็บไซต์ได้ เพื่อสร้างงานโดยใช้วิดเจ็ต (widgets) ที่จัดเตรียมให้

2. Blynk IoT ไลบรารี เป็นไลบรารีของคำสั่งเพิ่มเติมเพื่อพัฒนาโปรแกรมภาษา C/C++ สำหรับบอร์คไมโครคอนโทรลเลอร์เพื่อติคต่อกับ Blynk.Cloud เซิร์ฟเวอร์

 3. ฮาร์ดแวร์ ในที่นี้เป็นกล่องสมองกล WIO Terminal ที่ได้รับการโปรแกรมให้ทำงาน และติดต่อกับ Blynk เซิร์ฟเวอร์ รวมทั้งอุปกรณ์อินพุตเอาต์พุต

การทำงานในกรณีใช้งานกับสมาร์ตโฟนหรือแท็บเล็ต เมื่อมีการกดปุ่มบนหน้าจอของ Blynk app มันจะส่งข้อความคำสั่งผ่านอินเทอร์เน็ตไปยัง Blynk เซิร์ฟเวอร์ เพื่อส่งต่อไปยัง WIO Terminal เพื่อให้บอร์ดนำคำสั่งไปปฏิบัติตาม หรือทางกลับกันเมื่อมีการส่งค่าจาก WIO Terminal ไปยัง Blynk เซิร์ฟเวอร์เพื่อแสดงผลบนหน้าจอสมาร์ตโฟน

จากคำอธิบายเบื้องต้นพอจะอธิบายเป็นโครงสร้างได้ตามรูปที่ 11-4





11.3 ลงทะเบียนใช้งาน Blynk

11.3.1 ลงทะเบียนครั้งแรก

มีขั้นตอนดังนี้

(1) เชื่อมต่อคอมพิวเตอร์ที่ใช้ในการพัฒนาโปรแกรมเข้ากับเครือข่ายอินเทอร์เน็ต จากนั้นเปิด เว็บบราวเซอร์เพื่อเข้าไปยังเว็บเพจสำหรับเข้าสู่การใช้งานดังรูปที่ 11-5 ตามถิงก์ต่อไปนี้

https://blynk.cloud/dashboard/login

(2) คลิกที่ปุ่ม Create new account ดังรูปที่ 11-6

(3) จากนั้นจะปรากฏหน้าต่างสำหรับกรอกอีเมลที่ต้องการลงทะเบียนเพื่อสร้างบัญชีใช้งานกับ Blynk IoT ดังรูปที่ 11-7 ทำการระบุอีเมล (หมายเลข 1) แล้วคลิกเครื่องหมายถูกที่หัวข้อของการยอมรับ เงื่อนไขในการใช้งาน (หมายเลข 2) ตามด้วยคลิกปุ่ม Sign Up (หมายเลข 3) ไปยังขั้นตอนถัดไป

Dashboard	× +	v – 🗆 X	B Dashboard	× +	v - 0 X
← → C iii blynkd	toud/dashboard/login	- 영송 팩 🖪 😐 🗄	← → C iii blynk	c doud /dashboard/login	* @ ☆ 팩 🖪 🧧 :
	в			В	
	Log In			Log I	n
	EMAL			EMAIL	
	Please input your email			Please input your email	
	B			B	
	A password is required to log in			A password is required to log i	n
	Forgot password?			Forgot pass	word?
	Log in			Log In	
	Create new account			Create new a	ccount
		Privacy Policy			Privacy Policy

รูปที่ 11-5 หน้าต่าง login ของเว็บเพจ Blynk loT

รูปที่ 11-6 แสดงปุ่ม Create new account



กับ Blynk loT

รูปที่ 11-7 แสดงขั้นตอนการกรอกอีเมลลงทะเบียน รูปที่ 11-8 หน้าต่างแจ้งให้ผู้พัฒนาทำการยืนยัน การลงทะเบียน

(4) หน้าต่างแจ้งเตือนให้ผู้พัฒนายืนยันการลงทะเบียนด้วยอีเมลที่ลงทะเบียนไว้ก่อนหน้านี้ ดังรูปที่ 11-8

(5) จากนั้นให้ผู้พัฒนาเปิดอีเมลที่ได้รับจาก Blynk IoT คลิกที่ปุ่ม Create Password เพื่อกำหนด รหัสผ่านตามรูปที่ 11-9

(6) กำหนดรหัสผ่านอย่างน้อย 8 ตัว เมื่อกำหนดเรียบร้อย คลิกปุ่ม Next ตามรูปที่ 11-10

M Welcome to Blynk/Console - naki 🗙 🕂	v — I	
mail/u/2/?tab=km#inbox/FMtcgzGqQIIF/wGxGnMwImntzIgvIFIg	요 ☆ 팩 🖪	
Q ยังหาในอัเมล 💤	0 😣 🖩	в
← • • • • • • • • • • • • • • • • • • •		
В		Create Password Create a password which is hard to guess.
	0	
Welcome!		50-s0
We're excited to see you on board.		 Make it at least 8 symbols long
To get started, you'll need to create a password for your account.		Other tips:
Create Password The link will expire in 30 days.		Use uncommon words Use non-standard uPPercaSing Use creatif spellIIIIIing Use non-obviOu\$ number\$ & symbo1s
Thank you.		Log In Next

อีเมลยืนยันการลงทะเบียนจาก Blynk loT



ร**ูปที่ 11-11 แสดงขั้นตอนตั้งชื่อโปรไฟล์ของบัญชีผู้ใช้งาน Quickstart** เพื่อแนะนำรายละเอียุคต่าง (1) ตั้งชื่อโปรไฟล์ (2) คลิกปุ่ม Done

รูปที่ 11–9 เข้าสู่การกำหนดรหัสผ่านหลังจากได้รับ รูปที่ 11–10 แสดงขั้นตอนกำหนดรหัสผ่านของบัญชี ผู้ใช้งาน

> (1) กำหนดรหัสผ่าน (2) คลิกปุ่ม Next

(7) หน้าต่างสำหรับตั้งชื่อโปรไฟล์ ปรากฏขึ้นมา ผู้พัฒนาทำการกำหนดชื่อได้ ตามต้องการในหัวข้อ FIRST NAME จากนั้น คลิกปุ่ม Done ตามรูปที่ 11-11

(8) หน้าต่าง Blynk Tour ปรากฏ ู่ขึ้นมาเพื่อแนะนำรายละเอียดต่างเกี่ยวกับ Blynk IoT โดยในที่นี้จะข้ามขั้นตอนใน ส่วนนี้ไป โดยการคลิกปุ่ม Skip ตามรูปที่ 11-12

(9) หลังจากนั้น จะปรากฏหน้าต่าง เกี่ยวกับ Blynk IoT ในที่นี้จะข้ามขั้นตอน ในส่วนนี้ไปก่อน ด้วยการคลิกปุ่ม Cancel ตามรูปที่ 11-13



รูปที่ 11-12 หน้าต่างแนะนำรายละเอียดของ Blynk loT

Quickstart

This is a step by step guide to get your first device online and start controlling it from anywhere in the world in **less than 5 minutes**

What you will need:

- Supported hardware. Check the full list of supported hardware here.
- IDE. You can use Arduino IDE or PlatformIO or any other editor.
- Blynk Library
- It will be beneficial if you already know how to upload code to your hardware.

What we will do together:

- Create your first device in Blynk.Cloud
- Prepare firmware code and upload it to your device
- See your device online and control it from Blynk.Console and Blynk.Apps.





รูปที่ 11-13 แจ้งยกเลิกการรับชม Quick Start

(10) หน้าต่าง dashboard จะถูกแสดงขึ้นมา เมื่อการถงทะเบียนสำเร็จ โดยในหน้าต่างปัจจุบัน นี้แสดงหัวข้อ Billing เป็นการใช้งานแบบไม่มีค่าใช้จ่ายใดๆ (Free plan) ตามรูปที่ 11-14

📵 Blyn	k.Console × +					· · · · · ×	
\leftrightarrow \rightarrow	C B sgp1.blynk.cloud/dashboard/16124	/settings/billing				e 🖈 🗖 💿 :	
В Q	My organization - 56885W	Billing					
000	General Users	You are using	Free plan				
Ē	Locations Billing	USERS	DEVIC 1 oF5	es 0	of 2 🖉 Get 1	More	
Ţ	Tags						
n	ACCESS Roles and permissions	Plan Comparison Monthly 💽 Yearly (S	ave ~ 20%)				
	OEVELOPERS						
	Webhooks		Free	Plus	PRO	Business	
			\$0 /month Billed yearly	\$4.99 /month Billed yearly	\$42 /month Billed yearly	\$499 /month Billed yearly	
Ð				Devices	40 100	Web Only One App	
Ô					500	Two Apps	
0			Your current plan	🕼 Upgrade	🕫 Upgrade	Contact us	
2						Region: sgp1 Privacy Policy	

รูปที่ 11-14 แสดงหน้าต่าง dashboard ของ Blynk loT เมื่อการลงทะเบียนสำเร็จ

11.3.2 ขั้นตอนติดตั้งแอปพลิเคชั่น Blynk IoT บนระบบปฏิบัติการ iOS มีขั้นตอนดังนี้

(1)เข้าไปที่ AppStore ของระบบปฏิบัติการ iOS แล้วพิมพ์ค้นหาด้วยคำว่า Blynk IoT จากนั้น เลือกคาวน์โหลดแอปพลิเคชั่นตามรูปที่ 11-15 รองนกระทั่งการติดตั้งแอปพลิเคชั่นเสร็งสมบูรณ์

(2) ทดสอบเปิดแอปพลิเคชั่นขึ้นมา จะพบหน้าตาตามรูปที่ 11-16 ซึ่งเป็นหน้า Login เพื่อเข้า ใช้งาน โดยผู้พัฒนาสามารถใช้บัญชีหรือแอกเคาต์ที่เคยสมัครไว้ก่อนหน้านี้ล็อกอินใช้งานร่วมกับ Blynk IoT บนเว็บบราวเซอร์ได้ในเวลาเคียวกัน



Blynk loT บนระบบปฏิบัติการ iOS

ฐปที่ 11-15 หน้าตาแอปพลิเคชั่น ฐปที่ 11-16 แสดงหน้า Login ของ แอปพลิเคชั่น Blynk loT

11.4 ขั้นตอนการติดตั้งไลบรารีต่างๆ ที่จำเป็นในการสร้างโค้ดเพื่อติดต่อกับ Blynk IoT บนโปรแกรม Arduino IDE

สำหรับไลบรารี C/C++ ที่ผู้พัฒนาจำเป็นต้องติดตั้งเพิ่มเติมสำหรับในการพัฒนาโปรแกรมเพื่อ ให้ WIO Terminal ทำงานกับ Blynk IoT ได้ มีดังนี้

1. ใลบรารี Blynk

2. ใลบรารี ArduinoOTA

3. ไลบรารี ArduinoHttpClient

มีขั้นตอนดังนี้

(1) เปิดโปรแกรม Arduino IDE เลือกบอร์คพัฒนาเป็น Seeeduino WIO Terminal ตามรูปที่ 11-17

(2) เข้าสู่เมนู Maneger Libraries... เพื่อติดตั้งไลบรารีเพิ่มเติมทั้ง 3 ตัวที่กล่าวถึงไว้ข้างต้น ตามรูปที่ 11-18

(3) ที่หน้าต่าง Library Manager พิมพ์ค้นหาชื่อไลบรารีตัวแรกชื่อ Blynk ตามรูปที่ 11-19 เมื่อพบแล้ว คลิกที่ปุ่ม Install เพื่อติดตั้งไลบรารี โดยเลือกเวอร์ชั่นล่าสุดได้ทันที



รูปที่ 11-17 แสดงการตั้งค่าบอร์ดพัฒนาเป็น WIO Terminal



รูปที่ 11-18 แสดงการเข้าถึงเมนู Maneger Libraries...



รูปที่ 11-19 แสดงการเลือกติดตั้ง ไลบรารี Blynk



รูปที่ 11-20 แสดงการเลือกติดตั้ง ไลบรารี ArduinoOTA



รูปที่ 11-21 แสดงการติดตั้งไลบรารี ArduinoHttpClient

(4) ที่หน้าต่าง Library Manager พิมพ์ค้นหาชื่อไลบรารีตัวสองที่ชื่อ ArduinoOTA ตามรูป ที่ 11-20 เมื่อพบแล้วคลิกที่ปุ่ม Install เพื่อเริ่มการติดตั้งเช่นเดียวกัน

(5) ที่หน้าต่าง Library Manager พิมพ์ก้นหาชื่อไลบรารีตัวสุดท้ายที่ชื่อ ArduinoHttpClient ตามรูปที่ 11-21 เมื่อพบแล้วกลิกที่ปุ่ม Install เพื่อทำการติดตั้ง



11.5 ตัวอย่างการพัฒนาโปรแกรมเพื่อใช้งาน WIO Terminal กับ Blynk IoT

11.5.1 ตัวอย่างที่ 1 Light monitoring

เป็นตัวอย่างส่งค่าการทำงานของตัวตรวจจับแสงของ WIO Terminal ไปแสดงผลที่ Blynk IoT ้ผ่านเว็บบราวเซอร์และบนสมาร์ตโฟนอย่างง่าย สรุปเป็นไคอะแกรมตามรูปที่ 11-22 ส่วนขั้นตอนมีคังนี้

ส่วนที่ 1 การเตรียมการที่ Blynk IoT

(1) เชื่อมต่อคอมพิวเตอร์เข้ากับเครือข่ายอินเทอร์เน็ต แล้วเปิคเว็บบราวเซอร์เพื่อไปยังเว็บไซต์ https://blynk.cloud/dashboard/login จากนั้นถ็อกอินเพื่อเข้าใช้งาน

(2) คลิกเลือกเมนู Template ทางซ้ายมือ เพื่อเตรียมสร้างเทมเพลต (template) หรือแบบร่าง ขึ้นมาใช้งาน ตามรูปที่ 11-23

(3) หน้าต่าง Create New Template ปรากฏขึ้นตามรูปที่ 11-24 ทำการตั้งค่าดังนี้

o หัวข้อ NAME กำหนดชื่อเป็น Light

o หัวข้อ HARDWARE เลือกอุปกรณ์เป็น Seeed WIO Terminal

o หัวข้อ CONNECTION TYPE เลือกการเชื่อมต่อเป็น WiFi

o หัวข้อ DESCRIPTION สำหรับกำหนดคำอธิบายการทำงานตามที่ผู้พัฒนาต้องการ จากนั้นคลิกปุ่ม Done

Bysk.Conste						×
← → C # bly	k.da.d/dishboard/161241.products		ピ	Ŕ		ł
B						
9	0					
血	Start by creating your first	template				
1	Template is a digital model of a physical ubjet platform as a template to be assigned	d, it is used in Blyni d to devices.	6			
<i>4</i> 1	+ New Template					
0	2					
0						
2			Rep	0% 100	a	 Hic

รูปที่ 11-23 แสดงการเข้าเมนู Template ของ Blynk รูปที่ 11-24 แสดงหน้าต่าง Create New Template IOT

ใหม่

Blynki	Console × +	✓ - □ ×
\leftrightarrow \rightarrow	C blynk.cloud/dashboard/161241/products/edit/320810/info	여 년 ☆ 🖪 😐 🗄
В	Light	Cancel Save
Q	Info Metadata Datastreams Events Aut	omations Web Dashboard Mobile Dashboa 🔹
000	TEMPLATE NAME	TEMPLATE IMAGE (OPTIONAL)
Î	Light	G
	HARDWARE CONNECTION TYPE	
1	Seeed Wio Terminal V WiFi V	Add image
	DESCRIPTION	.png or .jpg, minimum width 500px
En	Get Light from Wio Terminal	
Ø		FIRMWARE CONFIGURATION #define BLYNK_TEMPLATE_ID "TMPLskjN361p"
0	27 / 128 TEMPLATE ID MANUFACTURER	#define BLYNK_DEVICE_NAME "Light"
	TMPLskjN361p My organization 56885W	top of your main firmware
ä		Region:sgp1 Privacy Policy

รูปที่ 11-25 แสดงหน้าต่างเทมเพลตที่ชื่อ Light ถูกสร้างขึ้นโดยหัวข้อ Info



รูปที่ 11-26 แสดงการวางวิดเจ็ต Gauge เพิ่มเข้าไปในโปรเจ็กต์

(4) หน้าต่างเทมเพลตที่ชื่อ Light ถูกสร้างขึ้นในหัวข้อ Info ตามรูปที่ 11-25 โดยแสดงรหัส ประจำตัวของเทมเพลตที่สำคัญ อันได้แก่

 o TEMPLATE_ID และ DEVICE_NAME ผู้พัฒนาจำเป็นต้องนำค่ารหัสทั้งสองของ เทมเพลตไปกำหนดไว้ที่ตอนต้นของโปรแกรมภาษา C/C++ เพื่อกำหนดการรับส่งข้อมูลระหว่าง คลาวด์เซิร์ฟเวอร์ของ Blynk IoT และ WIO Terminal

(5) คลิกเลือกที่หัวข้อ Web Dashboard เลือกวิคเจ็ต Gauge สำหรับแสดงค่าระดับแสง โดย การลากไปวางที่หน้าต่างโปรเจ็กต์ ตามรูปที่ 11-26

(6) ทำการกำหนดคุณสมบัติของวิคเจ็ต Gauge โดยการคลิกเมนูรูปฟันเฟืองที่วิคเจ็ต Gauge ตามรูปที่ 11-27



รูปที่ 11–27 แสดงเมนูรูปฟันเฟืองที่วิดเจ็ต Gauge สำหรับกำหนดคุณสมบัติ



B Byoka	ionosit x +	~ - D X
€ →	Byskabud;da/board/1524/products/dit/20010/de/boarc	+ i∂ 2) ⊡ (8 :
в	1 Labet	and the second second
Q	Gauge Settings	
	1(fut (OPhone)	
-	Light	
- El	Datastream	
1	You have no delastreams to select	1144
~	Analos	
t,G.	Virtual Pin	(-)
		0 100
0	2	
0		
10		
		Caroal Save
5		Report appl. Privacy Policy

รูปที่ 11-28 แสดงการกำหนดชื่อประจำตัววิดเจ็ต รูปที่ 11-29 แสดงการสร้าง Datastream แบบ Virtual Gauge สำหรับแสดงค่าความสว่าง

Pin

(7) จากนั้นหน้าต่าง Gauge Setting ปรากฏขึ้น ผู้พัฒนาสามารถกำหนดชื่อประจำตัววิดเจ็ต Gauge จากหัวข้อ TITLE ได้ตามต้องการ ในที่นี้กำหนดชื่อเป็น LIGHT ตามรูปที่ 11-28

(8) คลิกที่ปุ่ม Create Datastream เพื่อเพิ่มช่องทางการรับส่งข้อมูล แล้วเลือกรายการ Virtual Pin ตามรูปที่ 11-29

(9) จากนั้นหน้าต่างสำหรับกำหนดคุณสมบัติของ Virtual Pin จะปรากฏขึ้นตามรูปที่ 11-30 ทำการตั้งค่าดังนี้

o หัวข้อ NAME กำหนดชื่อเป็น Light

o ที่หัวข้อ ALIAS มีค่าเป็น Light

o หัวข้อ MAX (ค่าสูงสุด) เป็น 1023

o หัวข้อ DEFAULT VALUE เป็น 0

จากนั้น คลิกปุ่ม Create

(10) กลับมายังหน้าต่างหลักของ Gauge Setting ที่ถูกกำหนดคุณสมบัติไว้ก่อนหน้านี้ ตามรูป ที่ 11-31 โดยที่หัวข้อ Datastream ได้กำหนดเป็นช่อง V0 และขอบเขตของ Gauge จะถูกกำหนดการ แสดงผลอยู่ในช่วงตั้งแต่ 0 ถึง 1023 จากนั้นคลิกปุ่ม Save เพื่อบันทึกการตั้งค่า

(11) จากนั้นกลับมายังหน้าต่างหลักของ Web Dashboard ตามรูปที่ 11-32 คลิกปุ่ม Save เพื่อ บันทึกการตั้งค่าอีกครั้ง

🔋 Blynk.C	onsole × +				~	-		×
$\leftrightarrow \rightarrow 0$	B blynk.cloud/dashboard/161241/p	roducts/edit/320810/d			• ⊮			
B	Light				lancal		e	
Q	Gauge Settings							3
888	TITLE (OPTIONAL)						- 1	
000	Light							
	Datastream							
-11	Virtual Pin Datastream							
	NAME	ALIAS						
10		Light					- 1	
	PIN	DATA TYPE		Light			- 1	
	VO	Integer	V					
	UNITS							
	None		\sim	0 100				
	MIN MAX	D	EFAULT VALUE					
	0 102	3	0					
P	+ ADVANCED SETTINGS		_					
0		Cancel	Create					
				Cano	ы 📑	iave		
2					Reg	ion: sgp	1 Privac	

รูปที่ 11-30 แสดงหน้าต่างสำหรับกำหนดคุณสมบัติของ Virtual Pin

B Byrk Co	nuie × +	~ = 6 ×	📳 Byrk.Co	nak × +		v - D X
+ → c	B blyrk.doud/dathbcard/161241/products/edit/\$20110;dathboard	● 순 ☆ 🖪 🕒	+ > C	B blyrk.dbudfdahlbcard/161241/products	/edit/920010/diashboand	e e 🖈 🖪 💿 i
B	Light	- Cancel Serve	B	Light		Cancel Save
Q	Gauge Settings		Q	Info Metadata Datastreams	Events Automations Web	Dashboard Mobile
-	TITLE (DPTIONAL) Light		-	🖻 Widget Box	Device	
A	Dubuchranm		m	1 of 10 widgets	name	Show map Unotate
-	Light (V1)		-	Label 112	Device Company Owner Name	
	Override Datastream's Min/Max fields				Mag x Ø	
£2	Church color bared coursing	Light (vo)	00	Gauge	Dashboard	
		(333) 0 1023		æ	Las. 6 H. IDay 1W.	1N. 3 M. Cus.
				0 100	(i, (V0)	
£3			£3	Radial Gauge 🕢 🕢		
٥			0	42	0 1023,	
0			0			
2		Cancel Save	2	Chart		Region appl. Princy Policy

รูปที่ 11–31 แสดงหน้าต่างหลักของ Gauge Setting รูปที่ 11–32 แสดงการบันทึกหน้าต่างหลักของ Web ที่ได้รับการกำหนดคุณสมบัติไว้ก่อนหน้านี้ Dashboard ของเทมเพลตที่ชื่อ Light

ส่วนที่ 2 การเตรียมการสร้างโค้ดโปรแกรมภาษา C/C++ ที่ Arduino IDE

(12)เปิดโปรแกรม Arduino IDE ขึ้นมาเพื่อทำการพัฒนาโปรแกรมให้กับบอร์ด WIO Terminal (12A) สร้างโด้ดตามโปรแกรมที่ 11-1 โดย 2 บรรทัดแรกของโปรแกรมในส่วนของ BLYNK_TEMPLATE_ID และ BLYNK_DEVICE_NAME ผู้พัฒนาจำเป็นต้องกำหนดให้ตรงกับ เทมเพลตที่สร้างไว้ก่อนหน้านี้ โดยคัดลอกค่ารหัสดังกล่าวจากหัวข้อของ Template ตามรูปที่ 11-33



รูปที่ 11-33 แสดงค่ารหัส BLYNK_TEMPLATE_ID และ BLYNK_DEVICE_NAME ของเทมเพลตที่ชื่อ Light

#define BLYNK_TEMPLATE_ID "xxxxxxx"	// สำหรับกำหนดค่ารหัส TEMPLATE ID
#define BLYNK_DEVICE_NAME "xxxxx"	// สำหรับกำหนดค่ารหัส DEVICE NAME
#define BLYNK_FIRMWARE_VERSION "0.1.0"	// สำหรับกำหนดค่ารหัส FIRMWARE VERSION
#define BLYNK_PRINT Serial	
#define APP_DEBUG	
#include "BlynkEdgent.h"	// แนวกไลบรารีสำหรับติดต่อ Blynk loT
#include <tft_espi.h></tft_espi.h>	// ยนวกไลบรารีควบคุมการแสดงผล LCD

โปรแกรมที่ 11–1 ไฟล์ Blynk_Send_Light.ino โปรแกรมสำหรับทดสอบใช้งาน WIO Terminal อ่านค่าจาก ตัวตรวจจับแสงแล้วส่งขึ้นไปแสดงผลที่ Blynk IoT (มีต่อ)

```
// สร้างออปเจ็กต์ชื่อ tft จากคลาส TFT_eSPI สำหรับควบคุมการแสดงผล LCD
TFT eSPI tft;
                       // สร้างออปเจ็กต์ BlynkTimer ชื่อ timer
BlynkTimer timer;
                       // ฟังก์ชั่นสำหรับทำหน้าที่แสดงแลค่าระดับแสงและส่งไปแสดงแลที่ Blynk loT App
void sendLight()
                                                // อ่านค่าความเข้มของแสงจากตัวตรวจจับแสง
  int light = analogRead(WIO_LIGHT);
                                                // แปลงค่าระดับแสงเป็นข้อมูลสตริง
  String str = "Light: "+ String(light);
                                                // แสดงค่าระดับแสง
  tft.drawString(str+String(""), 80, 120);
                                                // ส่งค่าระดับแสงไปแสดงผลที่ Blynk App ผ่านช่อง VO
  Blynk.virtualWrite(VD, light);
void setup()
                                                //ใช้ Serial Monitor แจ้งสถานะการเชื่อมต่อเชิร์ฟเวอร์ Blynk
  Serial.begin(115200);
                                                // หน่วงเวลาเล็กน้อย
  delay(100);
                                                // กำหนดขาพอร์ตที่เชื่อมต่อกับตัวตรวจจับแสงให้เป็นอินพุด
  pinMode(WIO_LIGHT, INPUT);
                                                // เริ่มต้นการทำงานของจอแสดงแล
  tft.begin();
                                                // กำหนดทิศทางการหมุนหน้าจอแสดงแล
  tft.setRotation(3);
                                                // ระบายสีดำเป็นพื้นหลังหน้าจอ LCD
  tft.fillScreen(TFT_BLACK);
  tft.setTextColor(TFT_ORANGE,TFT_BLACK); // กำหนดสีตัวอักษรเป็นสีเขียว พื้นหลังสีดำ
                                                // กำหนดขนาดฟอนต์เป็น 3
  tft.setTextSize(3);
                                                // แสดงข้อความ Wait...
  tft.drawString("Wait...", 80, 120);
                                                // เริ่มต้นเชื่อมต่อไปยังเชิร์ฟเวอร์ Blynk IoT
  BlynkEdgent.begin();
                                                // เปิดใช้งานไทเมอร์อินเตอร์รัปต์ทุกๆ 1 วินาที
  timer.setInterval(1000L, sendLight);
                                                // กำหนดให้ไปทำงานที่ฟังก์ชั่น sendLight
}
void loop()
                                                // สั่งวัน Blvnk loT
  BlynkEdgent.run();
                                                // สั่งรันไทเมอร์
  timer.run();
}
```

การทำงานของโปรแกรม

หลังจากเชื่อมต่อไปยังเชิร์ฟเวอร์ Blynk IoT ได้สำเร็จ ผ่านทางการเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต โดย WIO Terminal ใช้โมดูล WiFi ติดต่อผ่านเราเตอร์เพื่อเข้าสู่เครือข่ายอินเทอร์เน็ตและ Blynk IoT App เราเตอร์หรือ สมาร์ตโฟนที่เป็นตัวกระจายสัญญาณ WiFi ตัวบอร์ด WIO Terminal จะเกิดอินเตอร์รัปต์จากไทเมอร์ทุกๆ 1 วินาที เพื่อกระโดดไปทำงานที่ฟังก์ชั่น sendLight โดยโปรแกรมจะอ่านค่าระดับแสงที่ตรวจวัดได้ไปแสดงผลที่หน้าจอ LCD และส่งไปยังเซิร์ฟเวอร์ Blynk IOT ผ่านช่องทางพอร์ตเสมือน VD

โปรแกรมที่ 11–1 ไฟล์ Blynk_Send_Light.ino โปรแกรมสำหรับทดสอบใช้งาน WIO Terminal อ่านค่าจาก ตัวตรวจจับแสงแล้วส่งขึ้นไปแสดงผลที่ Blynk IoT (จบ)

(12B) ผู้พัฒนาอาจเปิดไฟล์ของโปรแกรมตัวอย่าง (Example) จาก Arduino IDE ซึ่งมา พร้อมกับการติดตั้งไลบรารี Blynk เพื่อนำมาดัดแปลงเพิ่มเติม เป็นการช่วยอำนวยความสะดวกให้ พัฒนาโปรแกรมได้เร็วขึ้น มีขั้นตอนดังต่อไปนี้

(12B.1) ที่โปรแกรม Arduino IDE เลือกเมนู File>Examples>Blynk> Blynk.Edgent >Edgent_Wio_Terminal ตามรูปที่ 11-34



รูปที่ 11-34 แสดงเมนู File > Examples > Blynk > Blynk.Edgent > Edgent_Wio_Terminal



รูปที่ 11-35 แสดงไฟล์โปรแกรม Edgent_Wio_Terminal.ino บนโปรแกรม Arduino IDE

(12B.2) ใฟล์โปรแกรมตัวอย่างที่ชื่อ Edgent_Wio_Terminal.ino จะถูกเปิดขึ้นมา ตามรูปที่ 11-35

(12B.3) แนะนำให้ทำการ Save As เพื่อเก็บ ไฟล์ต้นทางไว้ แล้วตั้งชื่อ ไฟล์ใหม่ตามต้องการ

(13) เมื่อทำการสร้างโค้คหรือแก้ไขโค้คให้เป็นไปตามโปรแกรมที่ 11-1 แล้ว บันทึกไฟล์

(14) เชื่อมต่อ WIO Terminal เข้ากับคอมพิวเตอร์ผ่านทางพอร์ต USB จากนั้นทำการอัปโหลด โปรแกรมไปยัง WIO Terminal ให้เรียบร้อย

ส่วนที่ 3 เชื่อมต่อ Blynk IoT App กับ WIO Terminal และทดสอบการ ทำงาน

เป้าหมายของส่วนนี้คือ ทำการออกแบบสร้างแคชบอร์คบนเทมเพลตที่ตั้งก่าไว้กอนหน้านี้ และเชื่อมต่อกับ WIO Terminal รวมถึงการเข้าถึงเซิร์ฟเวอร์ Blynk IoT ในขณะที่โปรแกรมทำงาน สังเกตก่าระดับแสงที่ตรวจวัดได้บนจอแสดงผลของ WIO Terminal จากนั้นนำมือไปบังแสงที่ตก กระทบตัวตรวจจับที่อยู่บริเวณด้านหลังของ WIO Terminal สังเกตการเปลี่ยนแปลงก่าของระดับแสง ที่ตรวจจับได้ และสังเกตที่หน้าแคชบอร์คของ Blynk IoT App บนสมาร์ตโฟนและบนเว็บบราวเซอร์ กวรได้ผลลัพธ์ตรงกับที่แสดงบนจอแสดงผล WIO Terminal

สำหรับการออกแบบแคชบอร์คสำหรับ Blynk IoT App และทคสอบการทำงาน มีขั้นตอนคังนี้ (15) เปิค Blynk App บนสมาร์ต โฟนที่แนะนำการติคตั้งไว้ก่อนหน้านี้ขึ้นมา โดยในที่นี้ผู้เขียน ขอยกตัวอย่างจากการทำงานบนสมาร์ต โฟนที่ติคตั้งระบบปฏิบัติการ iOS ทำการถ็อกอินเข้าใช้งาน Blynk IOT ด้วยบัญชีที่เคยสมัครไว้ตามรูปที่ 11-36



รูปที่ 11-36 หน้าต่างล็อกอินขอ แอป Blynk IOT รูปที่ 11–37 แสดงปุ่ม Add New Device

Device



New Device เมื่อเริ่มต้นค้นหา การตรวจพบ WIO Terminal ที่รอ WIO Terminal

การเชื่อมต่อ

รายชื่อ Access Point ที่พบ เพื่อให้ WIO Terminal ทำการเชื่อมต่อเข้าสู่ เครือข่ายอินเทอร์เน็ต

(16) แตะที่ปุ่ม Add New Device ตามรูปที่ 11-37

(17) หน้าต่าง Add Device แสคงขึ้นมา แตะเลือกรายการ Connect to Wi-Fi ตามรูปที่ 11-38

(18) ที่หน้าต่าง Add New Device แตะที่ปุ่ม Start เพื่อเริ่มต้นก้นหาและเชื่อมต่อไปยัง WIO Terminal ตามรูปที่ 11-39

(19) เมื่อ Blynk App พบ WIO Terminal ที่กำลังรอการเชื่อมต่อจะปรากฏหน้าต่างแจ้งเตือน ตามรูปที่ 11-40 แตะปุ่ม Join เพื่อทำการเชื่อมต่อ

(20) รอจนกระทั่งปรากฏหน้าต่าง Choose Wi-Fi network ซึ่งแสดงรายชื่อเครือข่าย WiFi หรือ Access Point จากนั้นให้ผู้พัฒนาแตะเลือก Access Point ที่ต้องการให้ WIO Terminal เชื่อมต่อเข้าสู่ เครือข่ายอินเทอร์เน็ตตามรูปที่ 11-41 แตะปุ่ม Join เพื่อทำการเชื่อมต่อ



Setup

รูปที่ 11-42 แสดงหน้าต่าง Wi-Fi รูปที่ 11-43 หน้าต่าง Configuring รูปที่ 11-44 หน้าต่าง Device is Device แสดงสถานะการเชื่อมต่อ เครือข่ายอินเทอร์เน็ตของ WIO Terminal

Ready แสดงความพร้อมในการ เชื่อมต่อ

(21) จากนั้นจะปรากฏหน้าต่าง Wi-Fi Setup ของอุปกรณ์ Access Point ที่เลือก กรอกรหัส ้ผ่านในช่องด้านถ่างถัดมา ในกรณีที่ต้องการให้จำค่าการเชื่อมต่อของ Access Point นี้ให้แตะเลือก รายการ Remember the network for other devices จากนั้นแตะที่ปุ่ม Continue เพื่อไปยังขั้นตอน ต่อไป ตามรูปที่ 11-42

(22) หน้าต่าง Configuring Device ปรากฏขึ้นมา แสดงสถานะการเชื่อมต่อเครือข่ายอินเทอร์เน็ต ให้รอจนกระทั่ง WIO Terminal เชื่อมต่อกับอินเทอร์เน็ตเสร็จสมบูรณ์ดังรูปที่ 11-43

(23) หน้าต่าง Device is Ready ปรากฏขึ้นมาแสดงความพร้อมในการเชื่อมต่อตามรูปที่ 11-44 จากนั้นแตะที่ปุ่ม Continue



(24) จากนั้นปรากฏหน้าต่างป๊อบอัป Allow Notifications ให้แตะปุ่ม Skip เพื่อข้ามขั้นตอน การตั้งค่าแจ้งเตือนไปก่อน ตามรูปที่ 11-45

(24) แตะปุ่มเครื่องมือเพื่อเข้าถึงโหมดการออกแบบ ตามรูปที่ 11-46

(25) จากนั้นเข้าสู่โหมดการออกแบบของเทมเพลต Light ที่สร้างขึ้นก่อนหน้านี้ ทำการแตะ ที่เครื่องหมายบวก เพื่อเลือกวิดเจ็ตนำมาออกแบบหน้าตาของแอปพลิเคชั่นภายในเทมเพลตตามรูปที่ 11-47



(26) รายการวิดเจ็ต (widget) ปรากกฏขึ้นมา เลื่อนรายการ ไปทางค้านล่าง จนพบวิคเจ็ต Gauge ทำการแตะเลือก Gauge สำหรับแสดงค่าระดับแสงเข้าไปวางที่หน้าเทมเพลตตามรูปที่ 11-48

(27) จะได้วิดเจ็ต Gauge ที่เลือกมาใช้งานตามรูปที่ 11-49

(28) ทำการกำหนดคุณสมบัติของวิดเจ็ต Gauge โดยแตะที่ตัววิดเจ็ต จะปรากฏหน้าต่าง กำหนดคุณสมบัติขึ้นมา ในหัวข้อ DATASTREAM แตะที่ช่อง Select Datastream ตามรูปที่ 11-50

17:35 🕫		al) 🗢 😥 (17:36 🕫		at 🗢 👀	17:36 🖈		al 🗢 (4)
×	Choose	Clear	×	Gauge		×	Light	⊚ +
Light (v0)			Title (opti	onal)		-		
						vo		
			DATASTREAM]	~			
			FONT SIZE	RICT O AU	ТО			
			•	n				
			GAUGE COLOR					
				Delete				
					č		<u></u>	

Virtual Port สำหรับวิดเจ็ต Gauge เป็น v0

ถูกกำหนดใช้งาน Datastream ของวิดเจ็ต Gauge ในตัวอย่างนี้ ที่ช่อง Virtual Port

รูปที่ 11-51 แสดงการเลือกช่อง รูปที่ 11-52 แสดงวิดเจ็ต Gauge รูปที่ 11-53 ผลลัพธ์ที่สร้างขึ้น

(29) ปรากฏหน้าต่าง Choose... สำหรับเลือก Virtual Port ที่กำหนดข้อมูลจาก Web Dashboard ซึ่งกล่าวถึงก่อนหน้านี้ (จากส่วนที่ 1) เลือกรายการ Light(v0) ตามรูปที่ 11-51 ได้ผลลัพธ์ตามรูปที่ 11-52

โดยช่วงข้อมูลของ Gauge มีค่า 0 ถึง 1023 ตามที่เคยกำหนดไว้ภายใน Web Dashboard ทำการแตะที่ปุ่มเครื่องหมายกากบาทมุมบนค้านซ้ายเพื่อปิคหน้าต่างนี้ และกลับไปยังหน้าหลักของ การทำงานตามรูปที่ 11-53



รูปที่ 1 1–54 แสดงการทดสอบอ่านค่าระดับแสงของ WIO Terminal มีการแสดงผลการทำงาน ทั้งที่หน้าจอ แสดงผลของ WIO Terminal เอง และส่งค่ามาแสดงผลที่วิดเจ็ต Gauge ที่ประสานการทำงานกับ Blynk IoT

📵 Blyr	ik.Console × +		✓ – □ ×
÷⇒	C 🔒 blynk.cloud/dashboard/1	1241/global/filter/961791/devices	le 🖈 🖪 👩 i
B	My organization - 56885	My Devices	+ New Device
9	DEVICES	9	
200	My Devices All	1 Device name 2 Device owner © Status	Actions
		Blynk Light 1CA20 Nakhon Online	e
4ª	My locations All	0	
Ø	≗ USERS	à	
\odot	My organization members	1	
	All	1	
Do	With no devices	0	Region: sgp1 Privacy Policy

รูปที่ 11-55 แสดงการเข้าถึง Device ที่ทำการกำหนดไว้ก่อนหน้านี้ของ Blynk IOT



รูปที่ 1 1 – 56 ค่าระดับแสงจาก Wio Terminal ถูกส่งมาแสดงผลที่ Gauge บน Dashboard บนเว็บบราวเซอร์ ที่เคยออกแบบเอาไว้

(30) ที่หน้าต่างหลัก วิดเจ็ต Gauge จะแสดงค่าความสว่างทันที ดังรูปที่ 11-54

(31) จากนั้นทคสอบเข้าไปยัง Blynk IOT ในส่วนของเว็บบราวเซอร์ที่ล็อกอินอยู่ คลิกที่ปุ่ม ค้นหา Device จากนั้นที่หน้าต่างค้านขวามือจะปรากฏหน้าต่างหัวข้อ My Devices ขึ้นมา ให้เลือก เปิคตัวอุปกรณ์ที่ตั้งค่าไว้ก่อนหน้านี้ ซึ่งในที่นี้คือ อุปกรณ์ชื่อ Blynk Light xxxxx ตามรูปที่ 11-55

(32) หน้าต่าง **Dashboard** ของ Blynk IoT บนเว็บแสดงค่าระดับแสงที่ตรวจจับได้ผ่านทาง วิดเจ็ต **Gauge** ที่ออกแบบไว้ก่อนหน้านี้ (ส่วนที่ 1) ตามรูปที่ 11-56

11.5.2 ตัวอย่างที่ 2 IoT switch

ตัวอย่างถัดมาเป็นการจำลองการสั่งงานเปิด/ปิดไฟบ้านผ่านแอป Blynk IOT อย่างง่าย เป็นการ เชื่อมต่อกันระหว่าง WIO Terminal และ Blynk IOT เพื่อควบคุมอุปกรณ์เอาต์พุต โดยในตัวอย่างนี้ จะทำการแก้ไขวิดเจ็ตที่มีอยู่เดิมเพื่อให้ได้หน้าตาของ UI (User Intrerface : ส่วนติดต่อกับผู้ใช้งาน) ในการทำงานใหม่ โดยลบวิดเจ็ตเดิมออก แต่ยังคงใช้เทมเพลตเดิม

11.5.2.1 พัฒนาเว็บแดชบอร์ดของ Blynk IoT

เดิมที่ผู้พัฒนาใช้เทมเพลตที่มีวิดเจ็ต Gauge 1 ตัว ต้องลบออกเพื่อออกแบบ UI ใหม่ มีขั้นตอนดังนี้

(1) เข้าไปยังเมนู Template แล้วคลิกเลือกเทมเพลตที่ชื่อ Light ที่มีอยู่เคิม ตามรูปที่ 11-57

(2) เลือกหัวข้อ Web Dashboard แล้วคลิกปุ่ม Edit เพื่อเข้าไปแก้ไขตามรูปที่ 11-58

Blynk.Console × +	~ - 🗆 X
← → C 🔒 blynk.cloud/dashboard/161241/products	🖻 🕸 🖪 🦲 E
B Templates Q 1 Q Search Templates	+ New Template
Light 1 Device	
Æ3	

รูปที่ 11–57 แสดงเทมเพลตชื่อ Light ที่มีอยู่เดิม

Blynk.Co	nsole × +	~ - ¤ ×
← → 0	B blynk.cloud/dashboard/161241/product/320810/dashboard	2 e x 🛛 💿 :
В	Light 1	Duplicate Edit
Q	Info Metadata Datastreams Events Automations Web Dashboard	Mobile Dashboard
888		
(i)	Teg X 0	
-	Dashboard	
	Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months Custom	
	Light (V0)	
43		
Ø	290	
0	u 1023	
2		Region: sgp1 Privacy Policy

รูปที่ 11–58 แสดงหัวข้อ Web Dashboard

Blynk.Co	nsole × +		~ — Ш X
← → c	Bynk.cloud/dashboard/161241/products/	edit/320810/dashboard	e 🖈 🖪 😐 i
В	Light		Cancel Save And Apply
Q	Info Metadata Datastreams	Events Automations Web Dashboa	rd Mobile Dashboard 🛉 💿
888	Widget Box I of 30 widgets	name <u>A</u> Device Company Owner Name	2
	CONTROL	Tag × Ø	
-	Switch	Last H 6 Hours 1 Day 1 Week	1 Month 3 Mont_ Custom
ę9	Slider - (•) + 8		
Ø	Number Input	(357	
0		0 1023	
2	Image Button 🔒		Region: sgp1 Privacy Policy

รูปที่ 11–59 แสดงการลบ Gauge ออกจาก Template

Blynk.(Console × +						Ŷ	- 🗆 X
\leftrightarrow \rightarrow	C B blynk.cloud/dashboard/1	161241/product/320810/datastre	ams				É	x 🖪 👩 🗄
В	Light		1				Duplicate	Edit
Q	Info Metadata	Datastreams Events	Automations	Web Dashb	oard	Mobile Dashboard	2	
888	Q. Search datastream							
Ŵ	Id 🗄 Name	c Allas		a Color	Pin	章 Data Type 즉 또 Units	Is Raw 🔅	Min
-11	1 Light	Light			V0	Integer	false	0

รูปที่ 11-60 แสดงหัวข้อ Datastreams

(3) ใช้เมาส์ชี้ไปยังวิคเจ็ต Gauge คลิกเลือกไอคอนถังบยะเพื่อทำการลบวิคเจ็ต Gauge ออก จากนั้นคลิกปุ่ม Save And Apply ตามรูปที่ 11-59

(4) เลือกหัวข้อ Datastreams คลิกปุ่ม Edit เพื่อเข้าไปลบ Virtual Pin V0 ที่เคยตั้งค่าไว้ก่อน หน้านี้ ตามรูปที่ 11-60

(5) จากนั้นจะเข้าสู่โหมดการแก้ไขในส่วนของหัวข้อ Datastreams ใช้เมาส์ชี้ไปยังรายการ Light (การใช้งาน Virtual Pin V0) แล้วคลิกเลือกไอคอนถังขยะเพื่อลบการใช้งาน Virtual Pin V0 ออกจากระบบ ตามรูปที่ 11-61

(6) จากนั้นจะปรากฏหน้าต่างแจ้งเตือนว่า ต้องการแก้ไขข้อมูลการตั้งค่าในส่วนนี้หรือไม่ เพื่อ ป้องกันการเผลอลบโดยมิได้ตั้งใจ ตามรูปที่ 11-62

(7) ยืนยันการแก้ไขครั้งนี้ด้วยการกรอกข้อความ DELETE ในช่องด้านล่าง และคลิกเลือก รายการยืนยัน ตามด้วยการคลิกปุ่ม Delete ตามรูปที่ 11-63

(8) การตั้งค่าใช้งาน Virtual Pin V0 จะถูกลบออกจากหัวข้อ Datastreams ตามรูปที่ 11-64



รูปที่ 11–61 แสดงไอคอนถังขยะเพื่อลบการใช้งาน Virtual Pin VO



รูปที่ 11-63 แสดงขั้นตอนการยืนยันแก้ไขข้อมูล

รูปที่ 11-62 หน้าต่างแจ้งเตือนการแก้ไขข้อมูล



รูปที่ 11–64 แสดงให้เห็นว่า รายการ Virtual Pin VO ถูกลบออกไปจากหัวข้อ Datastreams แล้ว

(9) เลือกหัวข้อ Web Dashboard ตามรูปที่ 11-65

(10) ในหัวข้อ Widget Box เพิ่มวิคเจ็ต Switch เข้ามายัง Dashboard ตามรูปที่ 11-66



รูปที่ 11-65 แสดงหัวข้อ Web Dashboard



รูปที่ 11-66 แสดงการวางวิดเจ็ต Switch เพิ่มเข้ามาใน Dashboard

BlynkiC	ionsole x +				
→ (C i blynk.cloud/dashboard/161241/produc	ts/edit/320810/dashboard			e 🖈 🖪 💿 :
3	Light			*** Cancel	Save And Apply
2	Info Metadata Datastream	s Events Automations	Web Dashboard	Mobile Dashboard	0
909	🕾 Widget Box	Device na	ame Online		
1	1 of 30 widgets	Owner	Company Name	C Show map	UPGRADE
1	Switch	Teg X Ø			
		Deshboard +			
	Slider	Last Hour 6 Hours	1 Day 1 Week	1 Month 3 Months	Custom
3	- • 8				
9	Number Input	ð -			
9					
	O				

รูปที่ 11–67 แสดงไอคอนรูปฟันเฟืองสำหรับการตั้งค่าวิดเจ็ต Switch

(11) ใช้เมาส์ชี้ไปยังวิคเจ็ต Switch ที่หน้า Dashboard แล้วคลิกเลือกไอคอนรูปฟันเฟืองเพื่อ ทำการตั้งค่า ตามรูปที่ 11-67

(12) หน้าต่าง Switch Settings ปรากฏขึ้นมาสำหรับกำหนดการใช้งานวิดเจ็ต Switch สำหรับ LED2 ตามรูปที่ 11-68

(12.1) ที่หัวข้อหัวข้อ **TITLE (OPTIONAL)** กำหนดชื่อเป็น *LED2*

(12.2) กดปุ่ม Create Datastream แล้วเลือกรายการ Virtual Pin

📵 Blynk.C	onsole × +		×	-		×
$\leftrightarrow \rightarrow 0$	B blynk.cloud/dashboard/161241/products/edit/320810/dashboard		£	\$	C 🍈	:
B	Light			-	·····	
Q	Switch Settings				0	i.
888						
	Datastream					
-	You have no datastreams to select + Create Datastream 2 Digital Virtual Pin Enumerable	LED2				
€3	3					
0						
0			Cancel S	ave		
8			Regi	on:sgp1	Privacy	

รูปที่ 11-68 แสดงหน้าต่าง Switch Settings

Swit	ch Settings	S 🛛		รูบท 11–69 แสดงการดงคา Virtu ของวิดเจ็ต Switch ตัวแรกเป็น V2
TITLE (OF	PTIONAL)			
LED2				
Datast	tream			
Virtu	ual Pin Datastrea	m		
	NAME	ALIAS		
٥	Integer V2	Intege	r V2	
	201			
	V2		r VC	
	None			
	MIN	MAX	DEFAULT VALUE	
	-	_		
Swi	itch Settings	•		Press
Swi TITLE (C	tch Settings	0		Press
Swi TITLE (C LED2	tch Settings	•		Territoria
Swi TITLE (0 LED2 Datas	optional) 2. stream	•		Parent
Swi TITLE (C LED2 Datas Integ	optional) 2 stream ger V2 (V2)	OFFNALUE	V	
Swi TITLE (C LED2 Datas Integ ON VAL	optional) 2 stream ger V2 (V2)	OFF VALUE 0		
Swi TITLE (C LED2 Datas Integ ON VAL	optional) 2 stream ger V2 (V2)	OFF VALUE	· Ū	
Swi TITLE (C LED2 Datas Integ ON VAL	Show on/off labels	• OFF VALUE 0		LED2 (V2)
Swi TITLE (C LED2 Datas Integ ON VAL	Show on/off labels	OFF VALUE 0 0	 v v 	LED2 (V2)
Swi TITLE (C LED2 Datas Integ ON VAL 1	tch Settings	OFF VALUE 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	√ ♥	LED2 (V2)
Swi TITLE (C LED2 Datas Integ ON VAL 1	tch Settings	 OFF VALUE OFF VALUE O OFF LABEL OFF OFF (1) เลือก 	 I I	LED2 (M2) Com / off labels เพื่อแสดงสถานะ ON / O
Swi TITLE (C LED2 Datas Integ ON VAL 1	tch Settings	 OFF VALUE ○ ○	 Image: Show on LABEL กำหนดให 	LED2 (V2) (DED2 (DED2
Swi TITLE (C LED2 Datas Integ ON VAL 1	tch Settings	 Off VALUE 0 0 0 0 0 0 0 1 1 1 0 1 <l< td=""><td> อ ם ם ם ם ם</td><td>LED2 (M) DEF con/off labels เพื่อแสดงสถานะ ON/OL ห้แสดง ON สำหรับสถานะเปิด</td></l<>	 อ ם ם ם ם ם	LED2 (M) DEF con/off labels เพื่อแสดงสถานะ ON/OL ห้แสดง ON สำหรับสถานะเปิด
Swi TITLE (C LED2 Datas Integ ON VAL 1	tch Settings	 OFF VALUE 0 0 0 0 0 0 1 0 1 0 1 1 1 0 1 <l< td=""><td> Image: Show of the second seco</td><td>LED2 (VI) Ton/off labels เพื่อแสดงสถานะ ON/O ห้แสดง ON สำหรับสถานะเปิด ห้แสดง OFF สำหรับสถานะปิด</td></l<>	 Image: Show of the second seco	LED2 (VI) Ton/off labels เพื่อแสดงสถานะ ON/O ห้แสดง ON สำหรับสถานะเปิด ห้แสดง OFF สำหรับสถานะปิด

รูปที่ 11-70 การกำหนดสถานะต่าง ๆ ของวิดเจ็ต Switch

(13) จากนั้นจะปรากฏเมนูส่วนขยายตามรูปที่ 11-69 ในหัวข้อ PIN ตั้งค่าเป็น V2 จากนั้นกด ปุ่ม Create เพื่อยืนยัน

(14) จากนั้นกลับมายังหน้าหลักของ Switch Settings อีกครั้ง ทำการเลือก Enable รายการ Show on/off labels เพื่อแสดงสถานะ ON/OFF ของวิดเจ็ต Switch ตามรูปที่ 11-70 โดยตั้งก่าดังนี้

o หัวข้อ ON LABEL กำหนดให้แสดงข้อความ ON สำหรับสถานะ ON หรือเปิด

o หัวข้อ OFF LABEL กำหนดให้แสดงข้อความ OFF สำหรับสถานะ OFF หรือปิด

(15) เพิ่มวิดเจ็ต Switch เข้ามายัง Dashboard อีก 2 ตัว ตั้งชื่อเป็น LED1 ตั้งก่าใช้งานเป็น Virtual Pin V1 และ LED0 ตั้งก่าใช้งานเป็น Virtual Pin V0 ได้ผลลัพธ์ตามรูปที่ 11-71 จากนั้นกด ปุ่ม Save And Apply เพื่อยืนยันอีกครั้ง



รูปที่ 11-71 แสดงวิดเจ็ต Switch 2 ตัว, LED1 และ LEDO ที่ถูกเพิ่มเข้ามายัง Dashboard

11.5.2.2 พัฒนาโปรแกรมสำหรับ WIO Terminal บน Arduino IDE

(16) เปิดโปรแกรม Arduino IDE สร้างโค้ดตามโปรแกรมที่ 11-2 ตั้งชื่อไฟล์เป็น Blynk_IOT_LED_Demo.ino แล้วทำการอัปโหลดไปยัง WIO Terminal

สำหรับ 2 บรรทัดแรกของโปรแกรมในส่วนของ **BLYNK_TEMPLATE_ID** และ **BLYNK_DEVICE_NAME** ผู้พัฒนาจำเป็นต้องกำหนดให้ตรงกับเทมเพลตที่สร้างไว้ก่อนหน้านี้ โดย เข้าไปคัดลอกค่ารหัสดังกล่าวจากหัวข้อของเทมเพลตตามรูปที่ 11-26 (หัวข้อ 11.5.1 ก่อนหน้านี้)

```
// สำหรับกำหนดค่ารหัส TEMPLATE ID
#define BLYNK_TEMPLATE_ID "xxxxxxx"
                                                // สำหรับกำหนดค่ารหัส DEVICE NAME
#define BLYNK_DEVICE_NAME "xxxxx"
                                                // สำหรับกำหนดค่ารหัส FIRMWARF VERSION
#define BLYNK_FIRMWARE_VERSION "D.1.D"
#define BLYNK PRINT Serial
#define APP_DEBUG
                                                // แนวกไลบรารีสำหรับติดต่อ Blvnk IOT
#include "BlynkEdgent.h"
                                                // แนวกไลบรารีควบคุมการแสดงแล LCD
#include <TFT_eSPI.h>
                                 // สร้างออปเจ็กต์ชื่อ tft จากคลาส TFT_eSPI สำหรับควบคุมการแสดงผล
TFT eSPI tft;
                                  // สร้างออปเจ็กต์ BlynkTimer ชื่อ timer
BlynkTimer timer;
bool statusLed[]={false,false,false}; // กำหนดตัวแปรเก็บค่าสถานะการติด/ดับของ LED (true=ติด,false=ดับ)
                                  // ฟังก์ชั่นอัปเดทสถานะการติด/ดับของ LED แต่ละดวง
void updateStatus()
                                                    // แสดงข้อความพร้อมทำงาน
  tft.drawString("LED Demo ", 100, 20);
                                                    // LEDD ติดสว่าง ?
  if(statusLed[D])
                                                    // แสดงกราฟิก LEDO ติดสว่าง
      tft.fillCircle(240,140,30,TFT_GREEN);
  else
                                                    // แสดงกราฟิก LEDO ดับ
      tft.fillCircle(240,140,30,TFT_DARKGREY);
                                                    // IFD1 ติดสว่าง ?
  if(statusLed[1])
                                                    // แสดงกราฟิก I FD1 ติดสว่าง
      tft.fillCircle(160,140,30,TFT GREEN);
  else
                                                    // แสดงกราฟิก LED1 ดับ
      tft.fillCircle(160,140,30,TFT_DARKGREY);
                                                    // LED2 ติดสว่าง ?
  if(statusLed[2])
                                                    // แสดงกราฟิก LED2 ติดสว่าง
      tft.fillCircle(80,140,30,TFT_GREEN);
  else
                                                    // แสดงกราฟิก LED2 ดับ
      tft.fillCircle(80,140,30,TFT DARKGREY);
```

โปรแกรมที่ 11–2 ไฟล์ Blynk_loT_LED_Demo.ino โปรแกรมสำหรับทดสอบใช้งาน WIO Terminal ควบคุม LED ผ่านแอป Blynk loT (มีต่อ)

```
// ส่งข้อมูลสถานะการติด/ดับของ LEDO ไปยัง Blynk App
  Blynk.virtualWrite(V0, statusLed[0]);
                                                 // ส่งข้อมูลสถานะการติด/ดับของ LED1 ไปยัง Blynk App
  Blynk.virtualWrite(V1, statusLed[1]);
                                                 // ส่งข้อมูลสถานะการติด/ดับของ LED2 ไปยัง Blynk App
  Blynk.virtualWrite(V2, statusLed[2]);
}
void scanButton() // ฟังก์ชั่นตรวจจับการกด BUTTON A,B และ C
  if (digitalRead(WIO_KEY_A) == LOW)
                                                 // BUTTON A ถูกกด?
  {
                                                 // กลับสถานะ LEDO เมื่อ BUTTON A ถูกกด
       statusLed[0] = !statusLed[0];
                                                 // ปรับปรุงสถานะการติด/ดับของ LED แต่ละดวง
       updateStatus();
  }
  else if (digitalRead(WIO_KEY_B) == LOW)
                                                 // BUTTON B ถูกกด?
  {
                                                 // กลับสถานะ LED1 เมื่อ BUTTON A ถูกกด
       statusLed[1] = !statusLed[1];
                                                 // ปรับปรุงสถานะการติด/ดับของ LED แต่้ละดวง
       updateStatus();
  }
  else if (digitalRead(WIO_KEY_C) == LOW)
                                                 // BUTTON C ถูกกด?
  {
                                                 // กลับสถานะ LED2 เมื่อ BUTTON A ถูกกด
       statusLed[2] = !statusLed[2];
                                                 // ปรับปรุงสถานะการติด/ดับของ LED แต่ละดวง
       updateStatus();
  }
}
BLYNK_WRITE(V2) // Active for get data from Blynk app
{
  statusLed[2] = param[0].asInt();
                                                 // ปรับปรุงสถานะการติด/ดับของ LED แต่ละดวง
  updateStatus();
}
BLYNK_WRITE(V1)
  statusLed[1] = param[0].asInt();
                                                 // ปรับปรุงสถานะการติด/ดับของ LED แต่ละดวง
  updateStatus();
}
BLYNK_WRITE(VD)
  statusLed[0] = param[0].asInt();
                                                 // ปรับปรุงสถานะการติด/ดับของ LED แต่ละดวง
  updateStatus();
}
```

โปรแกรมที่ 11–2 ไฟล์ Blynk_loT_LED_Demo.ino โปรแกรมสำหรับทดสอบใช้งาน WIO Terminal ควบคุม LED ผ่านแอป Blynk loT (มีต่อ) void setup()

	pinMode(WIO_KEY_A, INPUT_PULLUP);	// กำหนดใช้งานพินที่เชื่อมต่อกับ BUTTON A เป็นอินพุต
	pinMode(WIO_KEY_B, INPUT_PULLUP);	// กำหนดใช้งานพินที่เชื่อมต่อกับ BUTTON B เป็นอินพุต
	pinMode(WIO_KEY_C, INPUT_PULLUP);	// กำหนดใช้งานพินที่เชื่อมต่อกับ BUTTON C เป็นอินพุต
	tft.begin();	// เริ่มต้นการทำงานของ LCD
	tft.setRotation(3);	// กำหนดทิศทางการหมุนหน้าจอของ LCD
	tft.fillScreen(TFT_BLACK);	// ระบายสีดำเป็นพื้นหลังหน้าจอ LCD
	tft.setTextColor(TFT_WHITE,TFT_BLACK);	// กำหนดสีตัวอักษรสีขาวพื้นหลังสีดำ
	tft.setTextSize(2);	// กำหนดขนาดฟอนต์เป็น 2
	tft.drawString("LED 0", 220, 80);	// ข้อความลาเบลประจำตัว LEDO
	tft.drawString("LED1", 140, 80);	// ข้อความลาเบลประจำตัว LED1
	tft.drawString("LED 2 ", 60 , 80);	// ข้อความลาเบลประจำตัว LED2
	tft.setTextColor(TFT_ORANGE,TFT_BLACK);	// กำหนดสีตัวอักษรสีส้มพื้นหลังสีดำ
	tft.setTextSize(3);	// กำหนดขนาดฟอนต์เป็น 3
	tft.drawString("Wait", 100, 20);	// แสดงข้อความแจ้งเตือนรอการเชื่อมต่อให้เรียบร้อย
	Serial.begin(115200); // กำหนดการใช้ Seria	ป Monitor ในกรณีที่แจ้งเดือนการเชื่อมต่อเซอร์ฟเวอร์ Blynk
	BlynkEdgent.begin(); // เริ่มต้นเชื่อมต่อไปยัง	มเซอร์ฟเวอร์ Blynk IOT
	timer.setInterval(500L, scanButton);	// เปิดใช้งานไทเมอร์อินเตอร์รัปต์ทุกๆ 0.5 วินาที
		// เรียกใช้งานไปยังฟังก์ชั้น scanButton
	updateStatus();	// ปรับปรุงสถานะการติด/ดับของ LED ในตอนเริ่มต้น
}		
/0	nd loop()	
	BlynkEdgent.run();	// สั้งรัน Blynk IoT
	timer.run();	// สังรันไทเมอร์

การทำงานของโปรแกรม

การทำงานของโปรแกรมหลักหลังจากเชื่อมต่กับเซิร์ฟเวอร์ Blynk ได้สำเร็จ จะเกิดอินเตอร์รัปต์จากไทเมอร์ ทุกๆ D.5 วินาที โปรแกรมจะกระโดดไปทำงานที่ฟังก์ชั่น scanButton ตามที่กำหนด เพื่อตรวจสอบการกด BUTTON A , B และ C ของ WIO Terminal พร้อมกับปรับปรุงการวาดสถานะของ LED แต่ละดวงที่จอแสดงผลเมื่อตรวจพบ ว่า BUTTON ถูกกด รวมถึงส่งข้อมูลสถานะไปปรับปรุงที่แอป Blynk IoT ด้วยการเรียกฟังก์ชั่น updateStatus ในกรณีที่ผู้ใช้งานกดปุ่ม BUTTON ตำแหน่ง LEDD ถึง LED2 ที่แอป Blynk IoT จะเป็นการส่งค่าสถานะการควบคุม กลับมายัง WIO Terminal ด้วยเช่นกัน

โปรแกรมที่ 11–2 ไฟล์ Blynk_loT_LED_Demo.ino โปรแกรมสำหรับทดสอบใช้งาน WIO Terminal ควบคุม LED ผ่านแอป Blynk loT (จบ)

11.5.2.3 ทดสอบการทำงาน

(ก) ทดสอบการทำงานเทียบกันระหว่าง WIO Terminal และแดชบอร์ด Blynk IoT

ในขณะที่โปรแกรมทำงานและ WIO Terminal เชื่อมต่อกับ Blynk IoT เรียบร้อยแล้ว ทำการ ทดสอบดังนี้

(1) ไปที่หน้า My Devices จากนั้นคลิกเลือกรายการอุปกรณ์ Blynk Light xxx (ในขณะนี้มี สถานะ Online โดยสังเกตที่ Status) ตามรูปที่ 11-72

(2) จากนั้นที่หน้า **Dashboard** วิคเจ็ต **Switch** ทั้ง 3 ตัวจะแสดงสถานะตรงกับ LED2 ถึง LED0 ที่ WIO Terminal ตามรูปที่ 11-73



รูปที่ 11–72 แสดงรายการอุปกรณ์ Blynk Light xxx เพื่อเข้าไปยังหน้า Dashboard

My o	rganization - \$6005W		×□	Blunk I	ight 1CA	20 Online				
$\leftarrow \textbf{Back}$			\bigcirc	& Nakhon	My orga	mization - 56	885W			
Search			-	Ø Add Tag						
		1.6	Dashboard	Timeline	Device I	nfo Met	ladata A	ctions Log		
1 Device		ψz	Latest	Last Hour	6 Hours	1 Day	1 Week	1 Month	3 Months	Custom
Bly	nk Light 1CA20									
1.000			LED2		LED1		LE	ED0		
				FF		ON		O ON		

รูปที่ 1 1 – 73 วิดเจ็ต Switch ทั้ง 3 ตัวที่หน้า Dashboard แสดงสถานะตรงกับ LED2 ถึง LEDO บนจอแสดงผล ของ WIO Terminal



(3) ทดสอบกดปุ่ม BUTTON A ของ WIO Terminal

รูป LED ที่ตำแหน่ง LED0 จะกลับสถานะการติด/ดับ (ติดเป็นสีเขียวและดับเป็นสีคำ) พร้อมกันนั้นสังเกตที่หน้าจอ Dashboard จะ ได้ผลลัพธ์ที่ LED0 ตรงกับจอแสดงผลของ WIO Terminal หรือ ไม่ (อาจต้องรอครู่หนึ่งเนื่องจากอาจมีช่วงหน่วงเวลาจากการทำงานของเซิร์ฟเวอร์ Blynk IoT อยู่บ้าง)

(4) กดวิดเจ็ต Switch ตำแหน่ง LED0 ของ Dashboard

กราฟิกรูป LED ที่ตำแหน่ง LED0 จะกลับสถานะอีกครั้ง โดยแสดงค้วยข้อความ ON/OFF

(5) ทำการทดสอบควบคุม LED1 และ LED2 เช่นเดียวกับข้อ 1 และ 2 ที่นำเสนอไปแล้ว

(ข) ทดสอบการทำงานเทียบกันระหว่าง WIO Terminal และแอป Blynk IoT บนสมาร์ตโฟน

(1) ที่สมาร์ต โฟนเปิด Blynk App ทำการล็อกอินเพื่อใช้งานจะปรากฏเทมเพลตที่เคยสร้างไว้ ก่อนหน้านี้ จากนั้นแตะปุ่มเครื่องมือเข้าสู่ Developer Mode เพื่อแก้ไข UI ตามรูปที่ 11-74



รูบท 11-77 แลดงการสบวดเจ Gauge ที่มีอยู่ภายในเทมเพลต รูปที่ 11–78 แสดงวิดเจ็ต Gauge ถูกลบออกจากเทมเพลต รูปที่ 11–79 ที่หน้าต่าง Widget Box แสดงการเลือกวิดเจ็ต Button

(2) แตะเลือก Template Light เพื่อเข้าไปแก้ไข UI ตามรูปที่ 11-75

(3) จากนั้นจะพบวิคเจ็ต Gauge ที่มีอยู่เคิม ให้แตะเลือกที่วิคเจ็ต Gauge ที่ต้องการลบตาม รูปที่ 11-76

(4) จะพบหน้าต่างกำหนดคุณสมบัติของ Gauge ให้แตะปุ่ม Delete เพื่อลบออกตามรูปที่ 11-77 จากนั้นออกจากหน้าต่างนี้ด้วยการกดปุ่มกากบาทที่มุมบนด้านซ้าย

(5) ที่หน้าต่างออกแบบหลัก (Developer Mode) จะเห็นว่า วิคเจ็ต Gauge จะถูกลบออกแล้ว แตะที่ปุ่มเครื่องหมายบวกเพื่อเพิ่มวิคเจ็ตตัวใหม่เข้ามา คังรูปที่ 11-78

(6) จากนั้นจะปรากฏรายการวิคเจ็ต ค้นหาวิคเจ็ต **Button** คังรูปที่ 11-79 ทำการแตะเลือกมา วางที่หน้าของเทมเพลต

13:16 🕇		al) 😤 😭	11:53 🕫		at 🗢 🚍	13:14 🕇		all 🗢 💼
~	Light	⊚ +	×	Button		~	Light	⊚ +
OFF			LED2	-1		V2		
				2				
			Integer V	2 (v2)	~			
			PU	SH SI	NITCH			
			ON ON	3 OFF OFF				
		861 867 M	FONT SIZE Mediu	m				
			TEXT COLOR					
ู รูปที่ 11-	-80 แสดงก	— ารวางวิดเจ็ต	รูปที่ 11	-81 แสดงก	ารกำหนด	ู รูปที่ 11-	82 แสดงวิด	- เเจ็ต Button

_ Button เพิ่มเข้าไปใน เทมเพลต คุณสมบัติของวิดเจ็ต Button

์ ที่กำหนดใช้งานอ้างอิงกับ LED2

(7) จากนั้นจะได้วิดเจ็ต Button มาวางบนเทมเพลตตามรูปที่ 11-80

(8) แตะที่วิคเจ็ต Button เพื่อเข้าไปกำหนคคุณสมบัติ จะปรากฏหน้าต่างสำหรับตั้งค่าตามรูปที่ 11-81 ทำการกำหนดคุณสมบัติดังนี้

o ตั้งชื่อ Button ตัวแรกเป็น LED2

o หัวข้อ DATASTREAM กำหนดเป็น V2

หัวข้อ MODE กำหนดเป็น SWITCH

แตะปุ่มกากบาทที่มุมบนด้านซ้าย เพื่อออกจากหน้าต่างตั้งค่า ได้ผลลัพธ์ตามรูปที่ 11-82

(9) ทำการเพิ่มวิคเจ็ต Button อีก 2 ตัวตามขั้นตอนก่อนหน้านี้

(9.1) ตั้งชื่อเป็น LED1 ตั้งค่าใช้งานเป็น V1

(9.2) ตั้งชื่อเป็น LED0 ตั้งค่าใช้งานเป็น V0 ตามรูปที่ 11-83

จากนั้นกลับไปยังหน้าหลักด้วยการแตะปุ่มถอยหลังที่มุมบนด้านซ้าย



รูปที่ 11–83 แสดงวิดเจ็ต Button ทั้ง 3 ตัวทำหน้าที่แสดงสถานะ LED2 ถึง LEDO ภายในเทมเพลต

LED2

V2



(10) ที่หน้าหลักของ Blynk App จะทำงานที่มีสถานะที่ตรงกับการแสดงผลของ WIO Terminal และ **Dashboard** ดังรูปที่ 11-84

(11) ผู้พัฒนาสามารถทคสอบปีค/เปิค LED แต่ละควงที่ Blynk App เพื่อทคสอบการเปลี่ยนแปลง สถานะว่าตรงกันหรือไม่ เช่นเดียวกับการทคสอบกับ **Dashboard** บนเว็บบราวเซอร์ก่อนหน้านี้

11.6 การใช้งาน Blynk IoT แบบระบุ ชื่อ Wi-Fi และ Token

ในการอธิบายตัวอย่างก่อนหน้านี้ทั้งหมด การเลือกเครือข่าย WiFi ให้ WIO Terminal เชื่อม ต่อเพื่อเข้าสู่เครือข่ายอินเทอร์เน็ตจะกระทำผ่าน Blynk IoT App ทั้งบนสมาร์ตโฟนหรือผ่านทางเว็บ ไซต์ ในหัวข้อนี้เป็นการนำเสนออีกวิธีหนึ่งที่ทำการเชื่อมต่อเครือข่าย WiFi โดยการระบุลงในโด้ดที่ อัปโหลดไปยัง WIO Terminal มีขั้นตอนดังนี้

(1) เชื่อมต่อคอมพิวเตอร์ที่ใช้พัฒนาโปรแกรมเข้ากับเครือข่ายอินเทอร์เน็ต เปิดเว็บบราวเซอร์ด เพื่อไปยังเว็บไซต์ของ Blynk IoT ทำการล็อกอิน สร้างเทมเพลตใหม่ โดยการคลิกที่ New Template ดังรูปที่ 11-85

(2) ตั้งชื่อเทมเพลตเป็น Lamp control เลือกอุปกรณ์เป็น Seeed Wio Terminal เลือกเชื่อม ต่อผ่าน Wi-Fi จากนั้นคลิกที่ปุ่ม Done ยืนยันการสร้างเทมเพลต ดังรูปที่ 11-86

(3) สร้างช่องทางการรับสั่งข้อมูล โดยเลือกไปที่เมนู Datastreams เพิ่มช่องทางการรับสั่งข้อมูล ด้วยการคลิกที่ปุ่ม New Datastream ดังรูปที่ 11-87

Blynk.Co	nsole	×		+						~	-		ł.	×
0 0 0	3	D	ŵ	blynk.cloud/dashboard/141589/products	É	Ø	⊿	D _r	9	۲	*		٦	Ξ
For quick acc	ess, place your book	marks h	ere c	n the bookmarks bar. Import bookmarks now								Other b	ookn	sarks
B														
Q														
000														
				Chart hu anatian un	Curt		1.1.							
				Start by creating you	r first te	emp	olate							
	Templat	e is a c	ligi	al model of a physical object. It is used in Bly	nk platfori	m as a	a templai	te to be a	ssigr	ned t	o de	vices.		
Ð				+ New Temp	late									
Ø														
\odot														
6														
ä										Regio	on: sgp	of Priva	icy Po	licy

รูปที่ 11–85 เริ่มต้นสร้างเทมเพลตใหม่ของ Blynk loT

Blynk.	Console		×	+											\sim	-			×
< ⊳	с	Д		blynk.clou	ud/dashboar	d/141589/produ	cts		ê	Ø	₽		o _r	1	Θ	*		٩	Ξ
For quick a	access, place your l	bookmarks l	here o	on the book	marks bar. Imp	ort bookmarks now	L										Other	r bookn	narks
В																٦			
Q		Creat	e N	New To	emplat	e										ł			
600 600		Lamp co	ntro	l												I			
Ŵ		Seeed W	/io Te	erminal		~	WiFi	ION TYPE						\sim		I			
		DESCRIPTIC	NN													de	vices	s.	
P		This is m	y ter	mplate												1			
0																J			
0											Ca	ncel	D	one					
2															Regio	n: sgp	1 Pri	vacy Po	olicy

รูปที่ 11-86 สร้างเทมเพลตใหม่ โดยเลือกฮาร์ดแวร์เป็น Seeed Wio Terminal



รูปที่ 11-87 สร้างช่องทางรับส่งข้อมูลใหม่สำหรับเทมเพลตชื่อ Lamp control



รูปที่ 11-88 เลือกช่องทางรับส่งข้อมูลผ่านขาพอร์ตเสมือนหรือ Virtual Pin

(4) เลือกกำหนดช่องทางรับส่งข้อมูลผ่าน Virtual Pin หรือขาพอร์ตอินพุตเอาต์พุตเสมือน โดยกลิกที่ + New Datastream แล้วเลือก Virtual Pin ดังรูปที่ 11-88

(5) ตั้งค่าการทำงานของช่องทางรับส่งข้อมูลเพื่อใช้ในการรับค่าจากตัวตรวจจับแสงของ WIO Terminal ดังรูปที่ 11-89 จากนั้นคลิกปุ่ม **Create** เพื่อยืนยัน

- o NAME กำหนดเป็น Light
- o PIN กำหนดเป็น V0
- o DATA TYPE กำหนดเป็น Integer
- o MAX ตั้งค่าเป็น 1023
- (6) สร้างช่องทางรับสั่งข้อมูลสำหรับรองรับก่าสถานะหลอดไฟหรือสวิตช์ ตั้งก่ารายละเอียดดังนี้
 - o NAME กำหนดเป็น Lamp
 - o PIN กำหนดเป็น *V1*
 - o DATA TYPE กำหนดเป็น Integer
 - o MAX ตั้งค่าเป็น 1
 - เมื่อตั้งค่าเสร็จแล้ว คลิกปุ่ม Create เพื่อยืนยัน คังรูปที่ 11-90

Blyn	k.Console		×	+							~		-		×
۵ Þ	с		D	blynk	.cloud/dashboard/141	589/products	/edit/509386/	6 0	1	De	1		* 1		Ξ
For quick	k access, place y	our bookr	narks here	e on the l	oookmarks bar. Import bo	okmarks now						I.	0	ther boo	kmarks
В	۲~		ontr								-	. 1	s	ave	
Q	Ir	Vir	tual	Pin [Datastream								hbo	ard 😡	
			NAME				ALIAS								
000		2	Light				Light								
			PIN				DATA TYPE								
	Datastr		VO			8	Integer			Ý			etry	, or ac	tuator
			UNITS												
đi.			None	5						Sý.					
Ø			MIN			MAX			DEFAULT VALUE						
Ŷ			0			1023			0				12		
0			⊕ AD'	VANCEI) SETTINGS				Cancel	Crea	te				
00	2										R	egion	:5gp1		Policy

Blynk.Console	£	× +				Ý	- 🗆 🗙
⊲ ⊵ c		🛱 🔒 blynk.clou	d/dashboard/141589/products	s/edit/509386/ 🖻 🕻	2	Q 🕲 Θ	* 🗆 🖻 🗄
For quick access, p	lace your bo	okmarks here on the bookm	arks bar. Import bookmarks now				Other bookmarks
B	L	oontrol				Concol	Save
Q	Ir V	irtual Pin Dat	astream				hboard 💿
		NAME		ALIAS			
000	ŧ	E Lamp		Lamp			
		PIN		DATA TYPE			atastream
Î	1	V1	×	Integer		(V)	
		UNITS					tions
4ª		None				×]	
0		MIN	MAX		DEFAULT VALUE		
		0	1		0		
Ø		ADVANCED SET	TINGS		Cancel	Create	
8					Carlot	oreand	

รูปที่ 11–90 หน้าต่างตั้งค่าช่องทางรับส่งข้อมูล ในที่นี้ใช้งานขาพอร์ตเสมือนหมายเลข 1 (V1) เพื่อขับ อุปกรณ์แสดงผล (Lamp) ของ WIO Terminal โดยมีข้อมูลเป็นเลขจำนวนเต็ม (Integer) 2 ตัวคือ 0 และ 1

В	Lamp control		***	Cancel Save
Q	Info Metadata Datastreams	Events Automations	Web Dashboard	Mobile Dashboard 💿
000	Q Search datastream			+ New Datastream
È	2 Datastreams			
P	☐ Id ‡ Name	\$ Alias	Color	Pin Actions
٢	🚝 1 Light	Light		VO
0	🚝 2 Lamp	Lamp		V1

รูปที่ 11–91 หน้าเว็บแสดงเทมเพลต Lam control ที่มีช่องทางรับส่งข้อมูลหรือ Datastreams 2 รายการคือ Light และ Lamp

(7) เมื่อสร้างเรียบร้อย จะมีช่องทางรับส่งข้อมูล (Datastreams) 2 รายการคือ Light และ Lamp ดังรูปที่ 11-91

(8) สร้างแผงควบคุมและแสดงผลหรือแดชบอร์ด โดยไปที่ Web Dashboard จากนั้นเลือกวิด เจ็ตเป็น Gauge สำหรับแสดงค่าที่อ่านได้จากตัวตรวจจับแสง โดยการลากมาวางดังรูปที่ 11-92

Lamp control	Cancel	Save
Info Metadata Dat	astreams Events Automations Web Dashboard Mobile Dashboard	
Widget Box	Device name online S Device Owner C Company Name Show map UPCRADE Tag X 0	
Gauge	Dashboard +	
42	Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months Custom	
0 100	Gauge	
Radial Gauge		
42	0 100	

รูปที่ 11–92 เริ่มต้นสร้างแดชบอร์ด โดยลากวิดเจ็ต Gauge มาวางบนแดชบอร์ด

в Lamp control Cancel Save Metadata Mobile Dashboard Info Datastreams Events Automations Web Dashboard 0 Q 200 😁 Widget Box Device name Online 1 of 30 widgets 0 C Show map 🙎 Device Owner 👔 Company Name 面 Tag × Ø -Gauge Dashboard 6 Hours 1 Day 1 Week 1 Month 3 Months Custom Last Hour 42 Gauge 0 100 q1 Radial Gauge G ٥ (\tilde{O}) 42 8 Region: sgp1 Privacy Poli

(9) ตั้งก่าวิดเจ็ตเพื่อใช้แสดงผลก่าแสงจากตัวตรวจจับ โดยกลิกที่รูปฟันเฟืองคังรูปที่ 11-93

(10) กำหนดชื่อของวิดเจ็ต Gauge เป็น *Light* และเลือก Datastream เป็น *Light (V0)* ดังรูปที่ 11-94 จากนั้นคลิกที่ปุ่ม Save เพื่อบันทึกการตั้งค่า

Lama control	- Cance C
Gauge Settings	
TITLE (OPTIONAL)	
Datastream Light (V0) V	
Override Datastream's Min/Max fields	714
Change color based on value	0 1023
-	
	Cancel Save

รูปที่ 11–94 กำหนดชื่อและช่องทางรับส่งข้อมูลของวิดเจ็ต Gauge

Info Metadata Datas	treams Events	Automations	Web Dashboard	Mobile Dashl	board		
皆 Widget Box		Device na	ame online				
2 of 30 widgets	Ø	A Device Ow	ner 📓 Company Na	me O	Show ma	UPCRADE	
CONTROL	Dashboard	+					
Switch							
	Last Hour	6 Hours	1 Day 1 Week	1 Month	3 Months	Custom	
Slider	Switch	6 Hours	1 Day 1 Week	1 Month	3 Months	Custom	
Slider +	8 Cast Hour	6 Hours	1Day 1Week	1 Month	3 Months	Custom	
Slider - · · · · · +	8 Switch	6 Hours ب	Light (vo)	1 Month 1	3 Months	Custom	
Slider - · · · · · · · · · · · · · · · · · · ·	8 Switch +	6 Hours	Light (vo) 0	1 Month	3 Months	Custom	

รูปที่ 11-95 เพิ่มวิดเจ็ต Switch บนพื้นที่สร้างแดชบอร์ด

(11) สร้างส่วนควบคุมการแสดงสถานะและเปิด-ปิดหลอดไฟ โดยเลือกวิดเจ็ต Switch เพื่อ กำหนดการแสดงสถานะของหลอดไฟ โดยลากมาวางบนพื้นที่สร้างแดชบอร์ดดังรูปที่ 11-95

(12) ตั้งชื่อวิดเจ็ต Switch เป็น *Lamp* และกำหนด Datastream เป็น *Lamp (V1)* ดังรูปที่ 11-96 จากนั้นคลิกปุ่ม Save เพื่อบันทึกการตั้งค่า

Switch Settin	ngs 💿	
TITLE (OPTIONAL)		
Lamp		
Datastream		
Lamp (V1)	~ D	
ON VALUE	OFF VALUE	Lamp (V1)
	a bala	
Show on/off la	abels	
 Show on/off la Hide widget n 	ame	
 Show on/off li Hide widget n 	ame	
 Show on/off li Hide widget n 	ame	Canad

รูปที่ 11–96 กำหนดชื่อและช่องทางรับส่งข้อมูลของวิดเจ็ต Switch

(13) เมื่อเพิ่มวิคเจ็ตที่ต้องการครบแล้ว คลิกที่ปุ่ม Save ดังรูปที่ 11-97 เพื่อบันทึกทั้งหมด เป็น อันเสร็จขั้นตอนการสร้างเทมเพลต

(14) ขั้นตอนต่อไปคือ ทำการเพิ่มอุปกรณ์ โดยไปที่รายการค้นหาดังรูปที่ 11-98 จากนั้นคลิกที่ ปุ่ม New Device

Info Metadata Datastream:	s Events Automations Web Dashboard Mobile Dashboard
음 Widget Box	Device name online
2 of 30 widgets	A Device Owner Company Name Show map UPCRADE
CONTROL	Teg X Ø
Switch	Dashboard +
	Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months Custom
Slider	Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months Custom
Slider - • + 8	Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months Custom
Slider - · · · · · · · · · · · · · · · · · · ·	Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months Custom
Slider - · · · · · · · · · · · · · · · · · · ·	Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months Custom
Slider - • + 8 Number Input - 0 + +	Last Hour 6 Hours 1 Day 1 Week 1 Month 3 Months Custom

รูปที่ 11-97 การบันทึกเทมเพลตของแดชบอร์ด Lamp control

Blymk	.Console	×	+												V.	-	0	Ō	×
₫ ▷	c	D	blynk.cloud/dashi	oard/141589/global	l/filter/861951			Ŀ	Ø	₽		-	15	•	*	≡J		3	Ξ
For quick	access, place your bo	ookmarks here	on the bookmarks bar. In	port bookmarks now												-	Other b	ookm	arks
B																			
_	My orga	nization -	5420OR																
Q	O DEVICES		Q																
898	My Device	25	0																
000	All		0			All	of you	r devi	ces	will	be h	iere.							
Ŵ			Q				You can	i activate n	ew dev	rices by	using								
A	My locatio	ons	0				yo	ar app for	IOS or	Androi	đ								
W	All		0				Download	for IOS		Þ	ownloa	d for Ar	droid						
P	O LICEDE		0			-			-										
a	My organi	tation me	mborr 1				2	+ Ner	v Devi	e.									
9	All	zacion me	1																
\odot	With no d	evices	1																

รูปที่ 11-98 คลิกรายการค้นหาเพื่อเตรียมเพิ่มอุปกรณ์

B	My organizati-	New Device			
000 000 000	My Devices All	Choose a way to create new de	vice		
1		From template	Scan QR code	Manual entry	
1	My locations All	000	00		aid
P	& USERS	000	(L).		olu
0	My organization				
0	With no devices			Cancel	
2					Region: sgp1 Privacy Policy

รูปที่ 11–99 เลือกเพิ่มอุปกรณ์จากเทมเพลต

(15) เลือกเพิ่มจากเทมเพลตที่ได้สร้างไว้ ดังรูปที่ 11-99

(16) เลือกเทมเพลตที่สร้างไว้ก่อนหน้านี้ นั่นคือ Lamp control จากนั้นคลิกปุ่ม Create เพื่อ ยืนยันการเพิ่มอุปกรณ์ ดังรูปที่ 11-100

(17) จากรูปที่ 11-101 จะเห็นว่า มีรายการอุปกรณ์เพิ่มขึ้นมา และอยู่ในสถานะ offline หรือปิด เพื่อรอการเชื่อมต่อ

(18) เลือกไปที่แท็บ Device Info เพื่อคัดลอกรหัสโทเคน (token) สำหรับนำไปใส่ในโค้ดที่เขียน ด้วย Arduino IDE ต่อไป ดังรูปที่ 11-102

	ing organization succo	124		
9	DEVICES	٩		
000	My Devices	0	New Device	
	All	0	Create new device by filling in the form below	vill be here.
<u>í</u>	CLOCATIONS	Q.	TEMPLATE	ces by using
1	My locations	0	Lamp control 🗸	oid
	All	0	DEVICE NAME	Download for Android
P	S USERS	a	Lamp control	
Ø	My organization members	1		
2	All	Ť	Cancel Create	
0	With no devices	1		

รูปที่ 11-100 ขั้นตอนการเพิ่มอุปกรณ์

Blynk	Console × +	✓ - □ X
$\triangleleft \triangleright$	C 🛛 blynk.cloud/dashboard	/141589/global/filter/861951/or 🖄 🛛 🗭 🧟 🧟 🛸 🗊 🗖 🖻 🗄
For quick	access, place your bookmarks here on the bookmarks bar. Impo	t bookmarks now
В		\times \square
Q	My organization - 54200R	Lamp control offline ••• & MyIOT M My organization - 5420OR
600 600	Search	O Add Tag
m	1 Device ↓ [^] _z	
		Late Last 6 Ho 1 Day 1 We 1 Mo 3 Mo Cust
2.		Lamp Light
٢		
\$		0
2		0 1023
		Region: sgp1 Privacy Policy

รูปที่ 11–101 แสดงการเพิ่มของอุปกรณ์ลงในเทมเพลต

В Q	My organization - 54200R	X I Lamp control offline A MyIOT My organization - 5420OR
000000000000000000000000000000000000000	Search	Dashboard Timeline Device Info Metadata Actions Log
Ŵ	1 Device ↓ ^A _z	
P	• Lamp control	#define BLYNK_TEMPLATE_ID "TMPL5ES_uyjh" #define BLYNK_DEVICE_NAME "Lamp control" #define BLYNK AUTH TOKEN "XkuKxGfZ_8zJpw6shRwcjrA3UH_ra1Xt"
٢		Template ID. Device Name and AuthToken should be declared at the very
ŝ		top of the firmware code.
		STATUS LAST UPDATED
2		Region: sgp1 Privacy Policy

รูปที่ 11-102 แสดงรหัส token ที่ทาง Blynk loT App สร้างขึ้นมาให้

```
#define BLYNK PRINT Serial
                                                 // คัดลอกจาก Device Info
#define BLYNK_TEMPLATE_ID "TMPL5ES_uyjh"
#define BLYNK_DEVICE_NAME "Lamp control"
                                                 // คัดลอกจาก Device Info
#define BLYNK AUTH TOKEN "XkuKxGfZ 8zJpW6shRWxjrA3IH ra1Xi"
                                                                        // คัดลอกจาก Device Info
                                                // ชื่อของตัวปล่อยสัญญาณ WiFi
char ssid[] = "xxxxxxxx";
                                                // รหัสย่านของตัวปล่อยสัญญาณ WiFi
char pass[] = "zzzzzzzz";
#include <rpcWiFi.h>
#include <WiFiClient.h>
                                                // ไลบรารีติดต่อกับ Blynk สำหรับ WIO Terminal
#include <BlynkSimpleWioTerminal.h>
#include <TFT eSPI.h>
TFT_eSPI tft;
BlynkTimer
              timer Sersor;
BlynkTimer
              timer_Button;
int statusLed = 0;
BLYNK WRITE(V1)
  int st = param.asInt();
  Serial.print("Got a value: ");
  Serial.println(st);
  if (st == 0)
  {
      statusLed = 0;
  }
  else
  {
      statusLed = 1;
  }
}
void scanButton ()
                                                    // ตรวจสอบการกดปุ่มของสวิตช์เข้ารหัส (สีน้ำเงิน)
  if (digitalRead(WIO_5S_PRESS) == LOW)
  {
                                                     // กลับสถานะตัวแปร
      statusLed = ! statusLed;
  }
                                                     // ตรวจสอบสถานะของกราฟิก LED
  if (statusLed == 1)
  {
                                                    // แสดงกราฟิก LED ติดสว่าง
      tft.fillCircle(160, 140, 30, TFT_GREEN);
  }
```

โปรแกรมที่ 1 1 – 3 ไฟล์ Blynk_Lamp_Control.ino สำหรับกำหนดให้ WIO Terminal วัดความเข้มแสงในสภาพ แวดล้อมนั้น แล้วส่งค่าไปแสดงยัง Blynk IoT และรับข้อมูลสั่งงานจาก Blynk IoT App มาควบคุมการติดดับ ของหลอดไฟจำลอง (มีต่อ)

```
else
  {
                                                       // แสดงกราฟิก LED ดับ
       tft.fillCircle(160, 140, 30, TFT_DARKGREY);
  }
}
void sendLight()
{
  int light = analogRead(WIO_LIGHT);
  String str = "Light: " + String(light);
  tft.drawString(str + String(" "), 80, 60);
                                                        // ส่งค่าที่อ่านได้จากตัวตรวจจับแสง
  Blynk.virtualWrite(VD, light);
                                                        // ส่งค่าสถานะหลอดไฟจำลอง
  Blynk.virtualWrite(V1, statusLed);
}
void setup()
{
  Serial.begin(9600);
  delay(100);
  pinMode(WIO_LIGHT, INPUT);
  pinMode(WIO_5S_PRESS, INPUT_PULLUP);
  tft.begin();
  tft.setRotation(3);
  tft.fillScreen(TFT_BLACK);
  tft.setTextColor(TFT_ORANGE, TFT_BLACK);
  tft.setTextSize(3);
  tft.drawString("Wait...", 80, 120);
                                                        // เรียกใช้ฟังก์ชัน sendLight ทุกๆ 1 วินาที
  Blynk.begin(BLYNK_AUTH_TOKEN, ssid, pass);
  timer_Sersor.setInterval(1000L, sendLight);
                                                        // เรียกใช้ฟังก์ชัน scanButton ทุกๆ 0.5 วินาที
  timer_Button.setInterval(500L, scanButton);
  tft.fillScreen(TFT BLACK);
}
void loop()
{
  Blynk.run();
  timer_Sersor.run();
  timer_Button.run();
}
```

โปรแกรมที่ 1 1 – 3 ไฟล์ Blynk_Lamp_Control.ino สำหรับกำหนดให้ WIO Terminal วัดความเข้มแสงในสภาพ แวดล้อมนั้น แล้วส่งค่าไปแสดงยัง Blynk IoT และรับข้อมูลสั่งงานจาก Blynk IoT App มาควบคุมการติดดับ ของหลอดไฟจำลอง (จบ) (19) ใปที่โปรแกรม Arduino IDE สร้างโค้คสำหรับโปรแกรม *Blynk_Lamp_Control* คังใน โปรแกรมที่ 11-3

(20) เชื่อมต่อ WIO Terminal เข้ากับพอร์ต USB ของคอมพิวเตอร์ แล้วอัปโหลดโปรแกรม

(21) เมื่ออัปโหลดโปรแกรมเสร็จ บอร์ด WIO Terminal จะทำงานทันที

ผลลัพธ์การทำงาน

WIO Terminal จะอ่านค่าจากตัวตรวจจับแสงและสถานะหลอดไฟจำลองในขณะนั้นมา แสดงที่หน้าจอแสดงผลของ WIO TErminal คังรูปที่ 11-103 และส่งไปยัง Blynk IoT คลาวค์เซิร์ฟเวอร์ ผ่านเครือข่ายอินเทอร์เน็ต ผู้ใช้งานสามารถสั่งเปิด-ปิคหลอดไฟจำลองด้วยการกดปุ่มสวิตช์สีน้ำเงิน หรือ Button บน WIO Terminal หรือบนแคชบอร์คที่หน้าเว็บของ Blynk IoT คลาวค์เซิร์ฟเวอร์ก็ได้ จะได้ผลลัพธ์การทำงานที่ตรงกัน



รูปที่ 11-103 ผลการทำงานของโปรแกรมที่ 11-3 ที่บอร์ด WIO Terminal

สำหรับการทำงานกับ Blynk IoT ของ WIO Terminal มีดังนี้

(1) รอการเชื่อมต่อ Blynk IoT คลาวค์เซิร์ฟเวอร์

(2) เมื่อเชื่อมต่อ ได้แล้ว จะแสดงค่าที่อ่าน ได้จากตัวตรวจจับแสง และสถานะหลอด ไฟดับ จำลอง เป็นวงกลมสีเทา

(3) เมื่อกดปุ่มสีน้ำเงินบน WIO Terminal จากวงกลมสีเทาที่หมายถึงดับหรือ OFF เปลี่ยน เป็นวงกลมสีเขียวที่หมายถึงทำงาน (ON)

ที่หน้าแคชบอร์คบนหน้าเว็บของ **Blynk IoT** จะแสดงสถานะ **Online** และการแสดงผล จะตรงกับหน้าจอแสดงผลของ WIO Terminal คังรูปที่ 11-104



รูปที่ 11–104 แดชบอร์ดของ Lamp control เมื่อทำงานและติดต่อกับ WIO Terminal เพื่ออ่านค่าจาก ตัวตรวจจับแสง และควบคุมการทำงานของหลอดไฟจำลอง

