บทที่ 2
ทฤษฎีและการศึกษางานวิจัยเกี่ยวข้อง

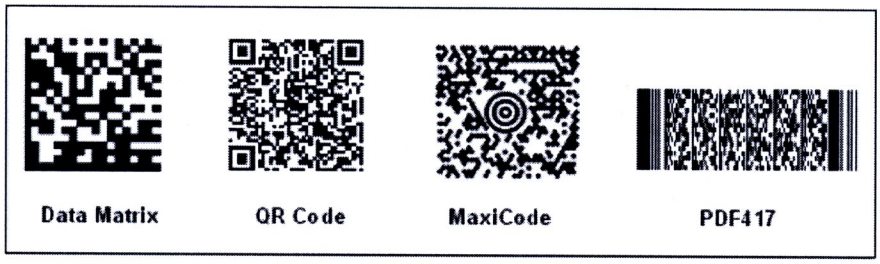
การพัฒนาโปรแกรมสำหรับรับ-ส่งคิวอาร์โค้ดของรูปภาพผ่านระบบเอสเอ็มเอสบนโทรศัพท์เคลื่อนที่ ผู้ศึกษาได้ค้นคว้าเอกสารและงานวิจัยที่เกี่ยวข้อง ซึ่งประกอบไปด้วย คิวอาร์โค้ด เอสเอ็มเอส เบส64 และงานวิจัยที่เกี่ยวข้อง

2.1 คิวอาร์โค้ด

ในปัจจุบันคิวอาร์โค้ด (QR Code: Quick Response Code) ถูกนำมาใช้อย่างกว้างขวางในงานด้านอุตสาหกรรม งานทางธุรกิจโฆษณา และการใช้งานส่วนบุคคล คิวอาร์โค้ดเป็นบาร์โค้ดชนิดหนึ่ง ที่พบเห็นได้ตามบนหนังสือนิตยาสาร หนังสือพิมพ์ ป้ายสินค้า กล่องหรือกระป๋องเครื่องดื่ม เป็นต้น คิวอาร์โค้ดสามารถอ่านได้โดยใช้โทรศัพท์เคลื่อนที่ ที่มีกล้องถ่ายภาพอยู่ในตัว โดยผ่านโปรแกรมการอ่านคิวอาร์โค้ด ก็สามารถแสดงข้อมูลข่าวสาร หรือเว็บไซต์ที่ซ่อนอยู่ในตัวคิวอาร์โค้ด

คิวอาร์โค้ด ซึ่งคิวอาร์โค้ดเป็นบาร์โค้ดสองมิติผลิตในประเทศญี่ปุ่นโดยบริษัทเดนโซ-เวฟ (Denso-Wave)ในปี 1994 โดยคิวอาร์โค้ด จัดอยู่ในกลุ่มของเมตริกซ์โค้ด(Matrix Code) หรือบาร์โค้ดสองมิติ (Two-Dimensional Bar Code)

ชัยกาล พิทยาเกษม และคณะ (2549) อธิบายไว้ว่า บาร์โค้ด 2 มิติ เป็นเทคโนโลยีที่ถูกพัฒนาเพิ่มเติมจากบาร์โค้ด 1 มิติ โดยที่บาร์โค้ด 2 มิติสามารถบรรจุข้อมูลได้ทั้งแนวตั้งและแนวนอน จึงสามารถบรรจุข้อมูลได้มากกว่าบาร์โค้ด 1 มิติ บาร์โค้ด 2 มิติสามารถถอดรหัสภาพได้แม้ภาพบาร์โค้ดบางส่วนมีการเสียหายอุปกรณ์ที่ใช้อ่านและถอดรหัสบาร์โค้ด 2 มิติ มีตั้งแต่เครื่องอ่านแบบซีซีดีหรือเครื่องอ่านแบบเลเซอร์เหมือนกับของบาร์โค้ด 1 มิติ จนถึงโทรศัพท์เคลื่อนที่แบบมีกล้องถ่ายรูปในตัวซึ่งต้องติดตั้งโปรแกรมถอดรหัสไว้ ในส่วนลักษณะของบาร์โค้ด 2 มิติ มีอยู่มากมายตามชนิดของบาร์โค้ด เช่น วงกลม สี่เหลี่ยมจัตุรัส หรือสี่เหลี่ยมผืนผ้าคล้ายกับบาร์โค้ด 1 มิติ ตัวอย่างบาร์โค้ด 2 มิติ ได้แก่ พีดีเอฟ417 แม็กซี่โค้ด ดาต้าเมตริกซ์ และ คิวอาร์ โค้ด ดังรูปที่ 2.1



รูปที่ 2.1 ตัวอย่างบาร์โค้ดแบบ 2 มิติ

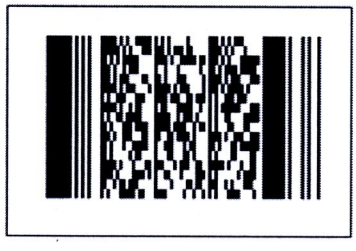
บาร์โค้ด 2 มิติ สามารถแบ่งออกเป็น 2 ประเภท

1) บาร์โค้ดแบบสแต็ก (Stacked Barcode)

2) บาร์โค้ดแบบเมตริกซ์ (Matrix Barcode)

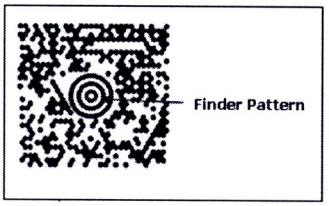
บาร์โค้ดแบบสแต็กมีลักษณะคล้ายกับการนำบาร์โค้ด 1 มิติมาวางซ้อนกันหลายแถว มีการทำงานโดยอ่านภาพบาร์โค้ดแล้วปรับความกว้างของบาร์โค้ดก่อนทำการถอดรหัส ซึ่งการปรับความกว้างนี้ทำให้สามารถถอดรหัสจากภาพที่เสียหายบางส่วนได้ส่วนที่เสียหายนั้นต้องไม่เสียหายเกินขีดจำกัดหนึ่งที่กำหนดไว้การอ่านบาร์โค้ด

แบบสแต็กสามารถอ่านได้ทิศทางเดียว เช่นอ่านจากทางซ้ายไปทางขวาหรือทางขวาไปซ้ายและอ่านจากบนลงล่างหรืออ่านจากล่างขึ้นบน ตัวอย่างบาร์โค้ดแบบสแต็ก เช่น บาร์โค้ดแบบพีดีเอฟ417

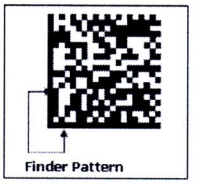


รูปที่ 2.2 ตัวอย่างบาร์โค้ดแบบพีดีเอฟ417

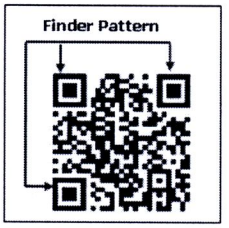
บาร์โค้ดแบบเมตริกซ์มีลักษณะหลากหลายและมีความเป็นสองมิติมากกว่าบาร์โค้ดแบบสแต็กที่เหมือนนำบาร์โค้ด 1 มิติไปว่างซ้อนกัน ลักษณะเด่นของบาร์โค้ดแบบเมตริกซ์คือมีรูปแบบค้นหา (Finder Pattern) ทำหน้าที่เป็นตัวอ้างอิงตำแหน่งในการอ่านและถอดรหัสข้อมูล ช่วยให้อ่านข้อมูลได้รวดเร็วและสามารถอ่านบาร์โค้ดเอียง หมุน หรือกลับหัว ตัวอย่างบาร์โค้ดแบบเมตริกซ์ เช่น แม็กซี่โค้ด ดาต้าเมตริกซ์ และ คิวอาร์โค้ด



ก) บาร์โค้ดแบบแม็กซี่โค้ด



ข) บาร์โค้ดแบบดาต้าเมตริกซ์



ค) บาร์โค้ดแบบ คิวอาร์ โค้ด

รูปที่ 2.3 กลุ่มบาร์โค้ดแบบเมตริกซ์

จากการที่ได้ศึกษาเกี่ยวกับบาร์โค้ด 2 มิติ ได้เห็นว่าบาร์โค้ดแบบ 2 มิติแต่ละแบบมีข้อแตกต่างกันในเรื่องของขนาดความจุของข้อมูล และบาร์โค้ดแบบคิวอาร์โค้ดเป็นบาร์โค้ดที่มีความจุของข้อมูลได้สูงไม่ว่าจะเป็นข้อมูลแบบตัวเลข ตัวอักษร เลขฐานสอง และตัวอักษรญี่ปุ่น ตามตารางที่ 2.1

ตารางที่ 2.1 การเปรียบเทียบบาร์โค้ด 2 มิติชนิดต่างๆ

บาร์โค้ด 2 มิติ		PDF417	MaxiCode	Data Matrix	QR Code
ผู้พัฒนา(ประเทศ)		Symbol Technologies (สหรัฐอเมริกา)	Oniplanar (สหรัฐอเมริกา)	RVSI Acuity CiMatrix (สหรัฐอเมริกา)	DENSO (ญี่ปุ่น)
ประเภทบาร์โค้ด		แบบสแต็ก	แบบเมตริกซ์	แบบเมตริกซ์	แบบเมตริกซ์
ขนาดความจุข้อมูล	ตัวเลข	2,710	138	3,116	7,089
	ตัวอักษร	1,850	93	2,355	4,296
	เลขฐานสอง (8 บิต)	1,018	-	1,556	2,953
	ตัวอักษรญี่ปุ่น	554	-	778	1,817
ลักษณะที่สำคัญ		- บรรจุข้อมูลได้มาก	- มีความเร็วในกรอ่านสูง	- บาร์โค้ดมีขนาดเล็ก	- บาร์โค้ดมีขนาดเล็ก - มีความเร็วในการอ่านสูง - บรรจุข้อมูลได้มาก
มาตรฐานที่ได้รับ		- ISO/IEC 15438 - AIM USS-PDF417	- ISO/IEC 16023 - ANIS/AIMBC10-ISS-MAxiCode	- ISO/IEC 16022 - ANIS/AIMBC11-ISS-DataMatrix	- ISO/IEC 18004 - JIS X 0510 - JEIDA-55 - AIM ITS/97/001ISS-QR Code

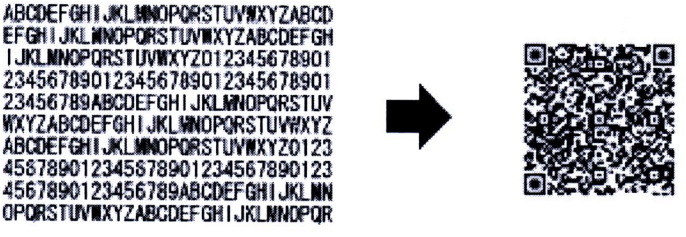
2.1.1 ขนาดความจุข้อมูลในคิวอาร์โค้ด

Denso-Wave (2012) อธิบายไว้ว่า ขนาดความจุข้อมูลในคิวอาร์โค้ดข้อมูลที่สามารถเก็บในรหัสคิวอาร์ขึ้นอยู่กับประเภทข้อมูล ในขณะที่บาร์โค้ดแบบเดิมมีความสามารถในการจัดเก็บได้สูงสุดประมาณ 20 หลัก รหัสคิวอาร์มีความสามารถในการจัดการข้อมูลทุกชนิดเช่นตัวอักษรตัวเลขและตัวอักษรคันจิ, คาตากานะ สัญลักษณ์ไบนารีและรหัสควบคุมสามารถเข้ารหัสในหนึ่งสัญลักษณ์ ในตารางที่ 2.2 เป็นการแสดงความจุของคิวอาร์โค้ดในแต่ละประเภทข้อมูล

ตารางที่ 2.2 ความจุข้อมูลรหัสคิวอาร์

ความจุข้อมูลรหัสคิวอาร์	
ตัวเลขอย่างเดี่ยว	สูงสุด 7,089 ตัวอักษร
ตัวอักษรและตัวเลข	สูงสุด 4,296 ตัวอักษร
ไบนารี (8 บิต)	สูงสุด 2,953 ไบต์
คันจิ/คาตากานะ	สูงสุด 1,817 ตัวอักษร

ในรูปที่ 2.4 เป็นตัวอย่างของการเข้ารหัสคิวอาร์โค้ดโดยข้อมูลที่เข้าเป็นข้อมูลประเภทตัวอักษรและตัวเลขที่มีขนาดความยาวของข้อมูล 300 ตัวอักษร



รูปที่ 2.4 รหัสคิวอาร์ ที่บรรจุข้อมูลตัวอักษรและตัวเลข 300 ตัวอักษร

2.1.2 โครงสร้างของคิวอาร์โค้ด

Tan Jin Soon (2008) กล่าวไว้ว่า โครงสร้างของคิวอาร์โค้ด คือสัญลักษณ์ประเภทเมทริกซ์ที่มีโครงสร้างเซลล์ที่จัดอยู่ในรูปแบบตาราง ประกอบด้วยรูปแบบการทำงานสำหรับการอ่านเป็นเรื่องง่ายและพื้นที่ข้อมูลที่จะถูกเก็บไว้ในรหัสคิวอาร์ โดยโครงสร้างของคิวอาร์โค้ด ดังรูปที่ 2.5 จะประกอบไปด้วย

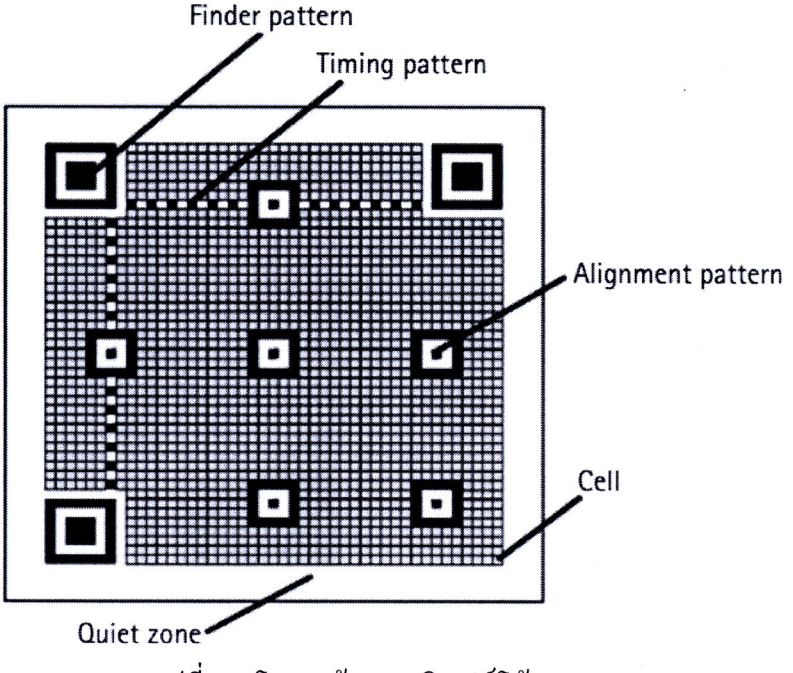
1) รูปแบบการหา (Finder pattern) คือรูปแบบสำหรับการตรวจหาตำแหน่งของรหัสคิวอาร์ โดยการจัดรูปแบบนี้ทั้งสามมุมของสัญลักษณ์ทำให้คิวอาร์โค้ดอ่านได้รอบทิศทาง(360 องศา) ความเร็วในการอ่านสูง

2) รูปแบบการควบคุม (Timing pattern) คือรูปแบบสำหรับการระบุพิกัดกลางของแต่ละเซลล์ในรหัสคิวอาร์ ที่มีรูปแบบสีดำและสีขาวจัดสลับกัน มันจะใช้สำหรับการแก้ไขพิกัดกลางของเซลล์ข้อมูลเมื่อสัญลักษณ์บิดเบี้ยวหรือเมื่อมีข้อผิดพลาด สำหรับเซลล์จะถูกจัดเรียงในทิศทางทั้งแนวตั้งและแนวนอน

3) รูปแบบการวางแนว (Alignment Pattern) คือรูปแบบสำหรับแก้ไขการบิดเบือนของรหัสคิวอาร์ ที่มีประสิทธิภาพสำหรับการแก้ไขการจัดตำแหน่งของรูปแบบจะถูกระบุในการแก้ไขความผิดเพี้ยนของสัญลักษณ์นี้เซลล์ที่แยกสีดำวางอยู่ในรูปแบบการจัดตำแหน่งเพื่อให้ง่ายในการตรวจสอบพิกัดกลางของรูปแบบการจัดตำแหน่ง

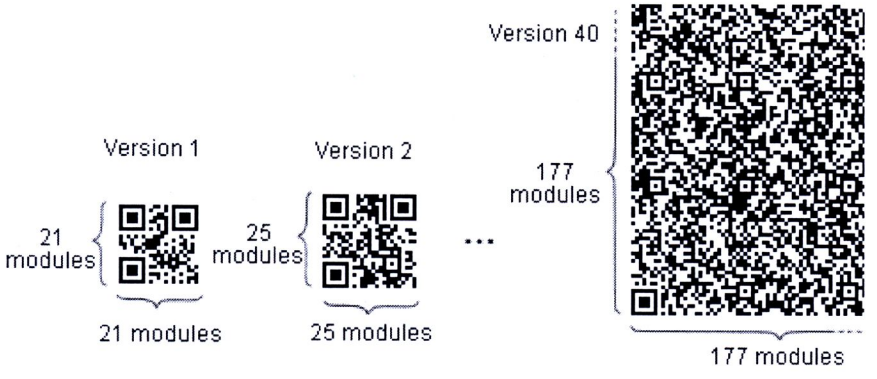
4) พื้นที่ว่าง (Quiet Zone) คือพื้นที่ขอบที่จำเป็นสำหรับการอ่านคิวอาร์โค้ด เพื่อทำให้ง่ายขึ้นที่จะมีสัญลักษณ์ที่ตรวจพบจากการอ่านโดยกล้อง

5) พื้นที่ข้อมูล (Data Area) คือพื้นที่ข้อมูลรหัสคิวอาร์จะถูกเก็บไว้(การเข้ารหัส) ลงในพื้นที่ข้อมูลจะถูกเข้ารหัสเป็นเลขฐานสองจาก 0 และ 1 อยู่บนพื้นฐานของกฎการเข้ารหัส รหัสจากเลขฐานสองจะถูกแปลงเป็นเซลล์สีดำและสีขาวแล้วจะได้รับการจัดพื้นที่ข้อมูลจะมีรหัสแบบรีดโซโลมอน (Reed Solomon Codes : RS) สำหรับข้อมูลที่จัดเก็บและการทำงานแก้ไขข้อผิดพลาด



รูปที่ 2.5 โครงสร้างของคิวอาร์โค้ด

การตั้งค่าโมดูล หมายถึงจำนวนของโมดูลที่มีอยู่ในสัญลักษณ์คิวอาร์ ในเวอร์ชันที่ 1 จะมีจำนวนโมดูลขนาด 21x21 โมดูล และในเวอร์ชันถัดๆไปจะมีขนาดมากขึ้นทุกๆ 4 โมดูล จนถึงเวอร์ชัน 40 จะมีจำนวนโมดูลขนาด 177x177 โมดูลในรูปที่ 2.6



รูปที่ 2.6 ให้เห็นความแตกต่างขนาดรหัสคิวอาร์โค้ดเวอร์ชัน1,2และ40

2.1.3 ระดับความผิดพลาดของคิวอาร์โค้ด

Denso-Wave (2012) อธิบายไว้ว่า ระดับความผิดพลาดของคิวอาร์ นั้นสามารถที่จะอ่านข้อมูลหรือกู้ข้อมูลได้แม้ว่าจะมีการฉีกขาดหรือมีคราบ สกปรกเพียงบางส่วน โดยสามารถกู้คืนได้มากที่สุด 30% ของรหัสตัวอักษร(CodeWord 1 Codeword= 8 bits) อนึ่งความมากน้อยของข้อมูลที่ถูกกู้คืนจะขึ้นอยู่กับความเสียหายที่เกิดขึ้น โดยแบ่งออกเป็น 4 ระดับ ดังนี้

- ระดับ แอล สามารถอ่านได้เมื่อมีความไม่สมบูรณ์ของบาร์โค้ด 7 เปอร์เซ็นต์
- ระดับ เอ็ม สามารถอ่านได้เมื่อมีความไม่สมบูรณ์ของบาร์โค้ด 15 เปอร์เซ็นต์
- ระดับ คิว สามารถอ่านได้เมื่อมีความไม่สมบูรณ์ของบาร์โค้ด 25 เปอร์เซ็นต์
- ระดับ เฮช สามารถอ่านได้เมื่อมีความไม่สมบูรณ์ของบาร์โค้ด 30 เปอร์เซ็นต์

ซึ่งระดับความผิดพลาดเหล่านี้เป็นระดับความผิดพลาดสากลที่ใช้ร่วมกันในทุกประเทศ เพื่อให้เข้าใจตรงกันว่าในระดับแอล ระดับเอ็ม ระดับคิว และระดับเฮช มีค่าของระดับความผิดพลาดเท่าไร

รูปที่ 2.7 สัญลักษณ์ของคิวอาร์โค้ดที่ลักษณะฉีกขาดหรือมีคราบบางส่วน

ในตารางที่ 2.3 เป็นการแสดงรายละเอียดของคิวอาร์โค้ดที่ผู้ศึกษาเลือกใช้งาน ในแต่ละเวอร์ชัน ซึ่งแสดงให้เห็นในส่วนหนึ่งของเวอร์ชันคิวอาร์โค้ด ตั้งแต่ เวอร์ชัน 15 ถึง 18 ของคิวอาร์โค้ด ส่วนการแสดงในแต่ละเวอร์ชันของคิวอาร์โค้ดสามารถดูได้จากภาคผนวก ก.

ตารางที่ 2.3 เวอร์ชันของคิวอาร์โค้ดในส่วนหนึ่ง

Version	Modules	ECC Level	Data bits	Numeric	Alphanumeric	Binary	Kanji
15	77x77	L	4,184	1,250	758	520	320
		M	3,320	991	600	412	254
		Q	2,360	703	426	292	180
		H	1,784	530	321	220	136
16	81x81	L	4,712	1,408	854	586	361
		M	3,624	1,082	656	450	277
		Q	2,600	775	470	322	198
		H	2,024	602	365	250	154
17	85x85	L	5,176	1,548	938	644	397
		M	4,056	1,212	734	504	310
		Q	2,936	876	531	364	224
		H	2,264	674	408	280	173
18	89x89	L	5,768	1,725	1,046	718	442
		M	4,504	1,346	816	560	345
		Q	3,176	948	574	394	243
		H	2,504	746	452	310	191

2.1.4 การเข้ารหัสคิวอาร์โค้ด

Thonky.com (2012) อธิบายไว้ว่า การเข้ารหัสของคิวอาร์ จะต้องทำการแปลงข้อมูลสายอักขระให้อยู่ในรูปของไบนารีของข้อมูลที่ต้องการ และจะมีข้อมูลเกี่ยวกับโหมดของการเข้ารหัสในรูปของไบนารีรวมไปกับข้อมูลที่ต้องการ ซึ่งมีขั้นตอนดังรูปที่ 2.8

สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุดงานวิจัย

รูปที่ 2.8 ลำดับขั้นตอนการเข้ารหัสคิวอาร์โค้ด

จากรูปที่ 2.8 เป็นลำดับขั้นตอนการเข้ารหัสคิวอาร์โค้ด เราสามารถอธิบายขั้นตอนการเข้ารหัสคิวอาร์โค้ดได้ดังนี้

1) เข้ารหัสโหมดการรับข้อมูลซึ่งในการเข้ารหัสของคิวอาร์โค้ดจะมีตัวบ่งชี้โหมดว่าข้อมูลที่รับเข้ามาเป็นกลุ่มไหน ตัวบ่งชี้โหมดจะเป็นบิตสตริงขนาด 4 บิต ดังแสดงในตารางที่ 2.4

สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุดงานวิจัย
วันที่ 26 พ.ย. 2555
เลขทะเบียน 250727

ตารางที่ 2.4 โหมดของการรับชนิดข้อมูลในคิวอาร์โค้ด

บิตสตริง	โหมดการรับข้อมูล
0001	โหมดตัวเลข
0010	โหมดตัวเลขและตัวอักษร
0100	โหมดเลขไบนารี
1000	โหมดภาษาญี่ปุ่น

ตัวอย่างการเข้ารหัส จะทำการเข้ารหัสข้อมูล “HELLO WORLD” ซึ่งข้อมูลเป็นโหมดตัวเลขและตัวอักษรจะมีตัวเลขไบนารีประจำโหมด **0010**

2) การเข้ารหัสความยาวของข้อมูล ในขั้นตอนนี้จะทำการกำหนดความยาวของชุดอักขระข้อมูลและแปลงเป็นไบนารี

ตัวอย่างการเข้ารหัส จะทำการเข้ารหัสข้อมูล “HELLO WORLD” ซึ่งมีจำนวนอักขระ 11 ตัวรวมอักษรเว้นวรรค เมื่อแปลงเป็นไบนารีคือ 1011 เมื่อเราเข้ารหัสความยาวของข้อมูลที่เราใช้งาน เราจะทำการสร้างคิวอาร์โค้ดตามรุ่นที่มี ซึ่งในแต่ละเวอร์ชันจะมีความต้องการของจำนวนบิตที่ต่างกัน ทั้งนี้ขึ้นอยู่กับข้อมูลที่จะถูกบรรจุลงในคิวอาร์โค้ดในแต่ละเวอร์ชัน

เวอร์ชันที่ 1 ถึง เวอร์ชันที่ 9

โหมดตัวเลข: ใช้ 10 บิต

โหมดตัวเลขและตัวอักษร: ใช้ 9 บิต

โหมดไบนารี: ใช้ 8 บิต

โหมดภาษาญี่ปุ่น: ใช้ 8 บิต

เวอร์ชันที่ 10 ถึง เวอร์ชันที่ 26

โหมดตัวเลข: ใช้ 12 บิต

โหมดตัวอักษรและตัวเลขที่: ใช้ 11 บิต

โหมดไบนารี: ใช้ 16 บิต

โหมดภาษาญี่ปุ่น: ใช้ 10 บิต

เวอร์ชันที่ 27 ถึง เวอร์ชันที่ 40

โหมดตัวเลข: ใช้ 14 บิต

โหมดตัวอักษรและตัวเลขที่: ใช้ 13 บิต

โหมดไบนารี: ใช้ 16 บิต

โหมดภาษาญี่ปุ่น: ใช้ 12 บิต

ซึ่งในตัวอย่างของคำว่า “HELLO WORLD” มีความยาวตังอักษร 11 ตัว แล้วเมื่อแปลงเป็นเลขไบนารีได้ 1011 จะเห็นได้ว่าข้อมูลที่ใส่เป็นโหมดตัวเลขและตัวอักษรจะอยู่ในช่วงของเวอร์ชันที่ 1 ถึงเวอร์ชัน 9 ต้องใช้จำนวน 9 บิต ซึ่งเราต้องทำการเติมบิต 0 ให้ครบจำนวนก็จะได้สายอักขระเป็น 000001011 เมื่อรวมกับโหมดรับข้อมูลจะได้ 0010 000001011

3) การเข้ารหัสข้อมูลเราจะจับคู่ตัวอักษร “HE LL O(space) WO RL D” ในการจับคู่ตัวอักษรเราจะใช้ค่ารหัสแอสกี้ของตัวอักษรตัวแรก คูณด้วย 45 แล้วนำมาบวกกับตัวอักษรตัวที่ 2 แล้วทำการแปลงค่าให้เป็นตัวเลขไบนารีขนาด 11 บิต ถ้ามีตัวอักษรตัวสุดท้ายเหลือเศษเราจะใช้ค่ารหัสแอสกี้มาแปลงเป็นตัวเลขไบนารีขนาด 6 บิต วิธีการทำแสดงดังรูปที่ 2.9

	HE	LL	O(space)	WO	RL	D
	(45*17)+14	(45*21)+21	(45*24)+36	(45*32)+24	(45*27)+21	13
	779	966	1116	1464	1236	13
0010 000001011	01100001011	01111000110	10001011100	10110111000	10011010100	001101

รูปที่ 2.9 การเข้ารหัสข้อมูล

4) จบบิตการเข้ารหัสเมื่อเราได้ข้อมูลแบบไบนารีมาแล้ว การสร้างต่อมาจะคำนึงถึงเวอร์ชันของคิวอาร์โค้ดและระดับการแก้ไขข้อผิดพลาดของคิวอาร์โค้ด ซึ่งในตัวอย่างเราจะใช้เวอร์ชัน 1 และระดับการแก้ไขข้อผิดพลาดอยู่ที่ระดับ คิว เมื่อดูตารางในภาคผนวก ก ประกอบจะเห็นได้ว่าจำนวนของข้อมูลจะมีได้ทั้งหมด 104 บิตถ้าข้อมูลของเรายาวน้อยกว่า 104 บิต จะทำการเพิ่ม 0 เข้าไปจำนวน 4 ตัว หรือถ้าเพิ่มเข้าไปแล้วเกิน 104 บิต ก็จะตัดออกเพื่อเอาความยาวแค่ 104 บิต

จำนวนไบนารีของเราจึงมี

0010 000001011 01100001011 01111000110 10001011100 10110111000 10011010100 001101 **0000**

5) จัดกลุ่มเป็น 8 บิต จะทำการแบ่งไบนารีออกเป็นกลุ่มกลุ่มละ 8 บิตถ้าข้อมูลชุดสุดท้ายไม่ครบ 8 บิตทำการใส่ 0 เข้าไปต่อท้ายข้อมูล

จำนวนไบนารีของเราจึงมี

00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000

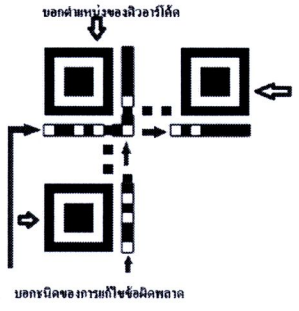
6) การเพิ่มคำต่อท้ายว่าข้อมูลสั้นเกินไปถ้าข้อมูลของเรายาวไม่พอครบ 104 บิต ยังมีชุดไบนารีพิเศษอีกสองชุดให้เราใส่จนครบความยาวที่ต้องการคือ 11101100 และ 00010001 ใส่ต่อกันไปจนครบ ซึ่งเรียกว่าบล็อกข้อมูล (1 บล็อกจะมี 8 บิต) ในความยาว 104 บิต นั้นจะใช้บล็อกข้อมูลทั้งหมด 13 บล็อก

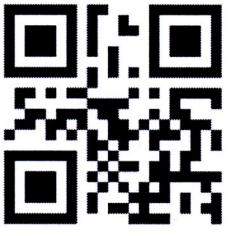
จำนวนไบนารีสุดท้ายของเราจึงมี

00100000 01011011 00001011 01111000 11010001 01110010 11011100 01001101 01000011 01000000 11101100 00010001 11101100

7) การรวมรหัสแก้ไขข้อผิดพลาดของข้อมูล ในรหัสคิวอาร์โค้ดนั้นข้อมูลบางส่วนของรหัสอาจหายไปบางส่วนแต่จะต้องสามารถอ่านข้อมูลได้ครบซึ่งจะใช้กระบวนการตรวจสอบความผิดพลาดรีด-โซโลมอน(Reed-Solomon error correction) (ดูวิธีการของรีด-โซโลมอนได้ในภาคผนวก ข) ขั้นตอนวิธีจะทำการดำเนินสร้างชุดข้อมูลแก้ไขข้อผิดพลาดในแต่ละบล็อกข้อมูลและทำการรวมชุดข้อมูลแก้ไขข้อผิดพลาดเอามาต่อท้ายข้อมูล

8) สร้างรูปแบบของรหัสจะทำการสร้างโครงสร้างเมตริกซ์ที่ว่างเปล่าที่มีขนาดที่ถูกต้องตามที่ระบุไวในตารางความจุเวอร์ชันของคิวอาร์โค้ด จากตัวอย่างของคำว่า “HELLO WORLD” เราต้องใช้ เวอร์ชัน 1 ซึ่งมีขนาด 21×21 ในรูปแบบจะทำการกำหนดรูปแบบการปรับตำแหน่งที่ใช้ในการอ้างอิงของคิวอาร์โค้ด, เพิ่มข้อมูลประเภทบอกเกี่ยวกับระดับของการแก้ไขข้อผิดพลาดแสดงในรูปที่ 2.10 ก. และบรรจุข้อมูลลงในคิวอาร์โค้ดแสดงในรูปที่ 2.10 ข.



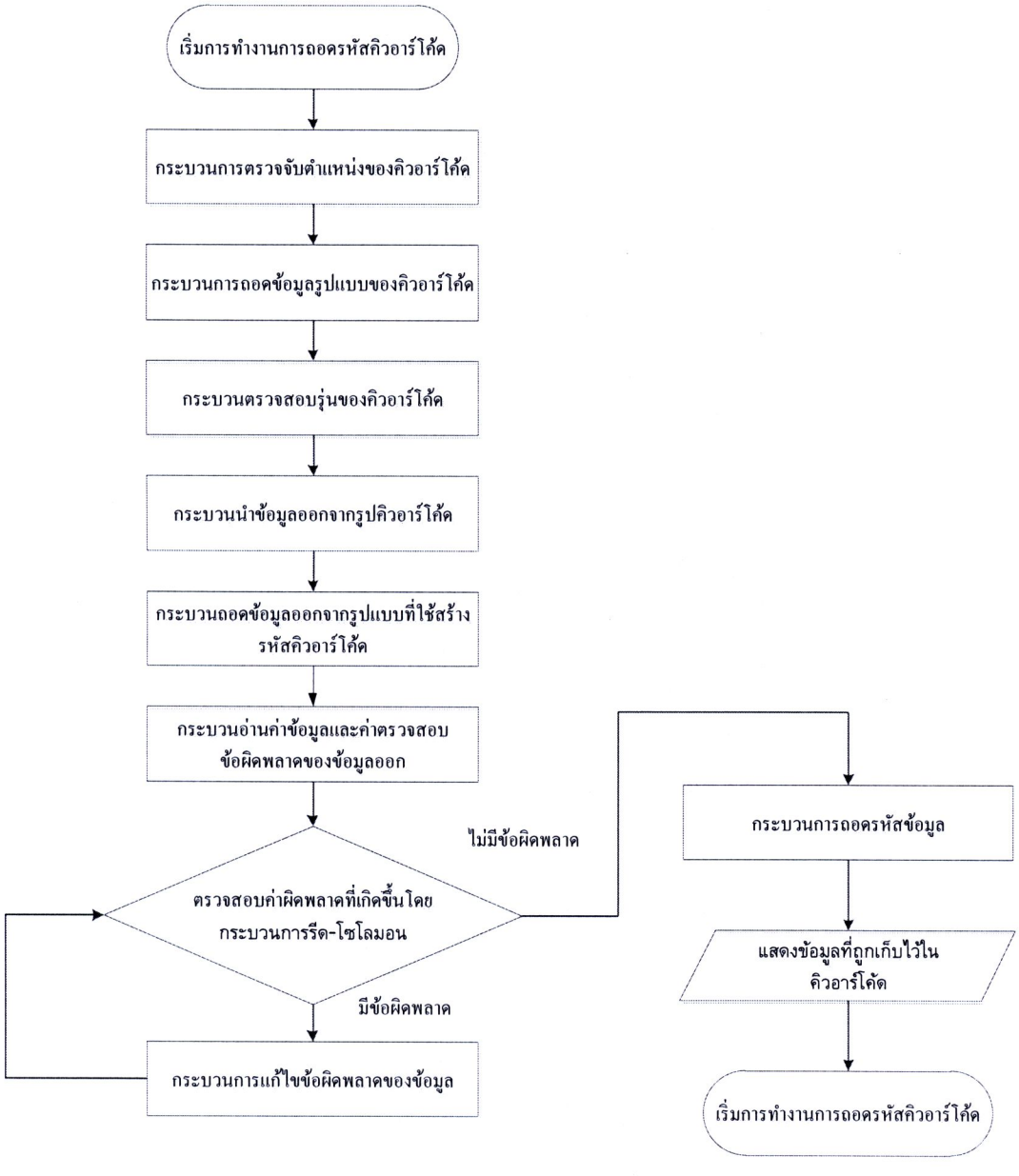


ก.ตำแหน่งของรูปแบบในคิวอาร์โค้ด ข.ข้อมูลที่ถูกบรรจุในคิวอาร์โค้ด

รูปที่ 2.10 ตัวอย่างในการเข้ารหัสคิวอาร์โค้ด

2.1.5 การถอดรหัสคิวอาร์โค้ด

International Standard ISO/IEC 18004 (2012) อิบายไว้ว่า การถอดรหัสคิวอาร์โค้ด จากการอ่านสัญลักษณ์คิวอาร์โค้ดให้ได้ข้อมูลออกมาเป็นสัญลักษณ์หรือตัวอักษรจากภาพถ่ายนั้นมีลำดับขั้นตอน ดังรูปที่ 2.11



รูปที่ 2.11 สำดับขั้นตอนการถอดรหัสคิวอาร์โค้ด

จากรูปที่ 2.11 สามารถอธิบายขั้นตอนการถอดรหัสคิวอาร์โค้ดได้ดังนี้

1) ระบุตำแหน่งของสัญลักษณ์คิวอาร์โค้ด และทำการจำแนกโมดูลที่มีสีขาวและดำ โดยที่สีขาวมีค่าเป็น 0 และสีดำมีค่าเป็น 1 มาเก็บไว้ในรูปแบบอาร์เรย์ และทำการระบุตำแหน่งคิวอาร์โค้ด โดยการหาสัญลักษณ์ของการระบบุตำแหน่งของคิวอาร์โค้ดทั้งสามมุม(Finder pattern) จากโมดูลที่ได้ทำการสร้างขึ้น
2) อ่านค่ารูปแบบของข้อมูลของคิวอาร์โค้ด จะทำการถอดส่วนที่ปกติดข้อมูลรูปแบบไว้ในคิวอาร์โค้ด และทำงานแก้ไขข้อผิดพลาดในโมดูลของรูปแบบคิวอาร์โค้ดตามความจำเป็นเช่น สัญลักษณ์ในคิวอาร์โค้ดมีความคลาดเคลื่อนก็จะทำการปรับแต่งให้ตรง ถ้าสำเร็จจะได้สัญลักษณ์คิวอาร์โค้ดแบบปกติ จะทำให้ทราบถึงระดับข้อผิดพลาดจากสัญลักษณ์คิวอาร์โค้ดที่สร้างขึ้นมา
3) อ่านรุ่นของคิวอาร์โค้ดจากสัญลักษณ์ที่ได้มาเพื่อใช้ในการอ้างอิงในการถอดรหัสข้อมูล
4) นำส่วนที่ปกปิดข้อมูลออกด้วยการ XORing ตามรูปแบบบิตที่ใช้ในการเข้ารหัสด้วยข้อมูลของรูปแบบของคิวอาร์โค้ด(mask pattern) ที่ได้จากการแยกจากรูปแบบช่องข้อมูล (format information)
5) อ่านตัวอักษรของสัญลักษณ์ต่างๆตามหลักเกณฑ์การวางตำแหน่งของรูปแบบและทำการกู้คืนข้อมูล (data) และส่วนของการตรวจสอบข้อผิดพลาด (error correction codewords) ของข้อความ
6) ตรวจหาข้อผิดพลาดด้วย error correction codewords ที่สอดคล้องกับระดับของข้อมูล ถ้าตรวจพบข้อผิดพลาดใดๆก็จะทำการแก้ไขให้ถูกต้อง
7) แยกข้อมูลที่ได้มาด้วยการแบ่งข้อมูลออกเป็นส่วนๆตามรูปแบบและจำนวนตัวอักษรที่กำหนดไว้
8) ถอดรหัสข้อมูลตัวอักษรให้เป็นไปตามรูปแบบต่างๆที่ใช้งานและแสดงผล

2.2 เอสเอ็มเอส (SMS: Short Message Service)

เอสเอ็มเอส หรือ การบริการส่งข้อความสั้น คือการส่งข้อความสั้น จากเครื่องโทรศัพท์เคลื่อนที่ จากเครื่องหนึ่ง ผ่านทางระบบของผู้ให้บริการโทรศัพท์เคลื่อนที่ไปยังโทรศัพท์เคลื่อนที่อีกเครื่องหนึ่ง หรือส่งผ่านจากระบบผู้ให้บริการส่งข้อความไปสู่เป้าหมายครั้งละหลายๆ เลขหมายในครั้งเดียว เอสเอ็มเอสได้ถูกสร้างมาครั้งแรกให้ทำงานร่วมกับโทรศัพท์เคลื่อนที่แบบดิจิตอล ระบบจีเอสเอ็ม(GSM) โดยข้อความแรกได้ถูกส่งในเดือนธันวาคม 1992 จากเครื่องคอมพิวเตอร์ส่วนบุคคลไปสู่เครื่องโทรศัพท์เคลื่อนที่บนโครงข่ายระบบจีเอสเอ็ม ของโวดาโฟน

(Vodafone) ในประเทศอังกฤษโดยสามารถส่งได้สูงสุด 160 ตัวอักษรต่อครั้งตามข้อกำหนดมาตรฐานขององค์การ อีทีเอสไอ (ETSI:European Telecommunications Standards Institute)

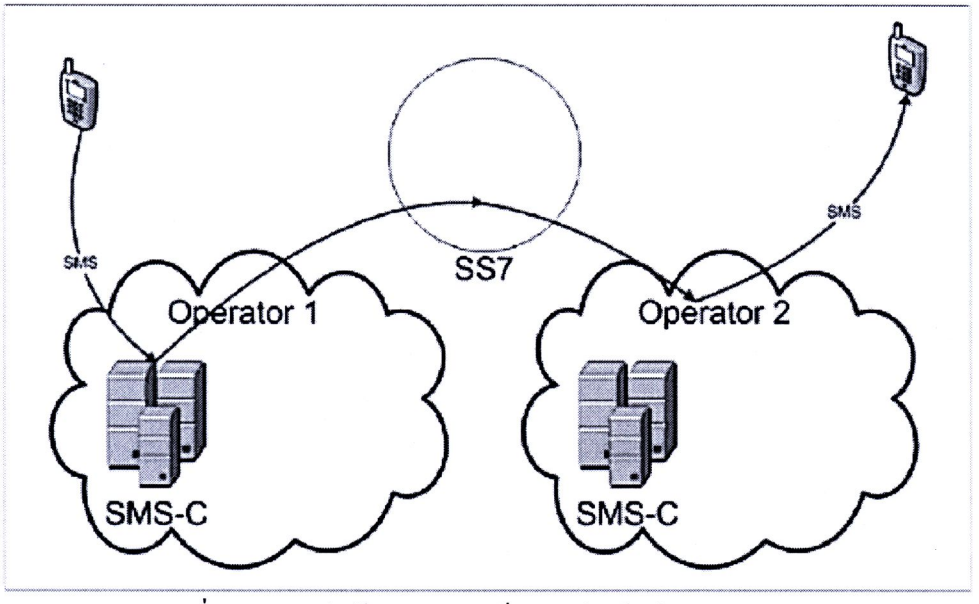
2.2.1 การทำงานของระบบรับส่งข้อความเอสเอ็มเอส

เกรียงศักดิ์และคณะ (2551) อธิบายไว้ว่า การทำงานของระบบรับส่งข้อความเอสเอ็มเอส มีหัวใจอยู่ที่ศูนย์ให้บริการเอสเอ็มเอส(SMSC:Short Message Service Center) ส่วนนี้จะเป็นเครื่องเซิร์ฟเวอร์ที่จะคอยเก็บข้อความที่ส่งมาจากหลายๆ ทางเช่น จากโทรศัพท์เคลื่อนที่ จากเว็บไซต์หรือจากผู้ให้บริการเอสเอ็มเอส ต่างๆ มารวบรวมเก็บเอาไว้เพื่อส่งต่อไปให้กับผู้รับ ศูนย์ให้บริการนี้จะทำการตรวจสอบว่าผู้รับอยู่ในพื้นที่ให้บริการของเซลล์ไซต์ใด หลังจากนั้นก็จะติดต่อไปยังโทรศัพท์เคลื่อนที่ของผู้รับเพื่อส่งเอสเอ็มเอส ที่อยู่ในคิวนี้ให้ และเมื่อผู้รับปลายทางได้รับข้อความนี้ก็จะลบข้อความที่เก็บไว้ในเครื่องเซิร์ฟเวอร์ของศูนย์ทิ้ง

ดังที่กล่าวมาข้างต้นว่าหัวใจของระบบเอสเอ็มเอสอยู่ที่ศูนย์ให้บริการเอสเอ็มเอส ทุกๆ ข้อความที่มีผู้ส่งไปให้กับผู้รับที่ปิดเครื่องไว้จะถูกเก็บไว้เป็นเวลา 24 ชั่วโมง เมื่อผู้รับเปิดเครื่องขึ้นมา ในทันทีที่ระบบตรวจพบก็จะส่งข้อความไปให้ นั้นก็รวมถึงกรณีที่อยู่นอกพื้นที่ให้บริการแล้วกลับเข้ามา ตัวอย่างเช่น มีผู้ส่งข้อความให้แต่ขณะนั้นผู้รับอยู่ในพื้นที่ไม่มีสัญญาณ ทันทีที่กลับออกมาจากจุดนั้น ระบบก็จะส่งข้อความให้ทันที หรือแม้แต่โทรศัพท์เคลื่อนที่เก็บข้อความไวจนเต็ม ไม่สามารถรับข้อความเพิ่มได้ เมื่อลบข้อความเก่าทิ้ง ระบบก็จะส่งข้อความใหม่ที่รออยู่ในคิวให้ทันที จะเห็นได้ว่าเอสเอ็มเอสเป็นบริการที่จะไม่ทำให้พลาดการติดต่อได้เลย

เอสเอ็มเอสคือข่าวสารที่ส่งให้กับโทรศัพท์เคลื่อนที่และถือว่าเป็นอุตสาหกรรมที่นิยมใช้กันมากในปัจจุบันและก่อให้เกิดข่าวสารเกิดจากตัวผู้ใช้โทรศัพท์เคลื่อนที่เองหรือเกิดจากโอเปอร์เรเตอร์ของผู้ให้บริการ การส่งข่าวสารที่ได้รับจะมีรูปแบบที่แตกต่างกัน บริการเอสเอ็มเอลเป็นเพียงโปโตคอล อินพุต เอาท์พุต สำหรับแอพพิเคชั่นซึ่งมีประโยชน์และรู้จักกันมาก เช่น บริการการโหวต บริการส่งข้อความข่าวสาร บริการค้นหา บริการโต้ตอบ

รูปแบบพื้นฐานของการส่งข้อความเป็นเพียงการส่งให้คนต่อคนและจะสามารถส่งข้อความได้ไม่เกิน 160 ตัวอักษรในภาษาอังกฤษและตัวเลข และการส่งเอสเอ็มเอสเป็นภาษาไทยนั้นใน 1 ข้อความจะส่งได้ไม่เกิน 70 ตัวอักษร (รวมสระและเว้นวรรค) เมื่อเราส่งข้อความจากโทรศัพท์เคลื่อนที่ ที่เราตั้งค่าเอาไว้ในซิมการ์ดและเมื่อส่งไปจะผ่านทางศูนย์ผู้ให้บริการส่งไปยังผู้รับโดยใช้โปโตคอล เอสเอส7 (SS7 Protocol) และผู้ให้บริการเก็บข้อความไว้จนกระทั่งส่งข้อความให้ผู้รับได้ ดังแสดงในรูปที่ 2.12



รูปที่ 2.12 การส่งข้อความเอสเอ็มเอสผ่านโปโตรคอล เอสเอส7

เนื้อหาที่ใหญ่กว่าเช่น ข้อความแบบต่อเนื่อง หลายข้อความ หรือข้อความที่ยาว สามารถถูกส่งโดยใช้หลายข้อความ โดยที่แต่ละข้อความจะมีข้อมูลของผู้ใช้(UDH:User Data Header) ประกอบด้วย เนื่องด้วยข้อมูลของผู้ใช้ถูกบรรจุอยู่ในส่วนของข้อความที่ถูกส่งออกไป จำนวนของอักขระที่ส่งต่อส่วนอยู่ต่ำกว่า 153 ตัวอักษรสำหรับการเข้ารหัสที่ 7 บิต, 133 ตัวอักษรสำหรับการเข้ารหัสที่ 8 บิต และ 67 ตัวอักษรสำหรับการเข้ารหัสที่ 16 บิต โทรศัพท์มือถือที่ได้รับข้อความจะรวบรวมข้อความทั้งหมดและนำเสนอแก่ผู้ใช้โดยรวมเป็นหนึ่งข้อความยาวๆ ขณะที่ทฤษฎีมาตรฐานอนุญาตให้สูงสุด 255 ส่วน

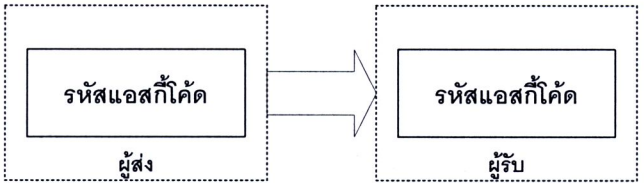
2.2.2 โหมดของการรับส่งข้อมูลเอสเอ็มเอส

เอกสรรค์ มลาเช็ค และ ฐาปกิตติ์ ไค่นุ่นน้อย (2549) อธิบายไว้ว่า การรับส่งข้อมูลเอสเอ็มเอส มีอยู่ด้วยกัน 2 โหมด คือ เท็กซ์โหมด (Text Mode) และ พีดียูโหมด (PDU Mode: Protocol Description Unit Mode) การส่งข้อความในเท็กซ์โหมดนั้นจะเป็นการนำข้อความที่ต้องการส่งมาเข้ารหัสก่อนแล้วค่อยส่งข้อมูลในพีดียูโหมดอีกที อย่างไรก็ตามในมือถือบางรุ่นอาจไม่สนับสนุนการใช้งานในเท็กซ์โหมด ซึ่งการเข้ารหัส (ส่ง) และถอดรหัส (รับ) สำหรับในเท็กซ์โหมดนี้มีหลายแบบด้วยกัน เช่น พีซีซีพี437(PCCP437) พีซีดีเอ็น(PCDN) 8859-1 ไออาร์เอ(IRA) และจีเอสเอ็ม (GSM)

การเชื่อมต่อกับมือถือเพื่อรับส่งข้อความสามารถเลือกใช้ได้ทั้ง 2 โหมด แต่จะเห็นได้ว่าการเลือกใช้เท็กซ์โหมดจะมีข้อจำกัดทั้งจากการที่มือถือบางรุ่นอาจไม่สนับสนุนและยังถูกจำกัดด้วย

วิธีการเข้าและถอดรหัส ซึ่งมีเพียงไม่กี่แบบตามที่กล่าวมาข้างต้น แต่ถ้าเลือกพีดียูโหมดจะสามารถเลือกหรือสร้างการเข้ารหัสและถอดรหัสได้ทุกรูปแบบตามต้องการโดยไม่มีข้อจำกัด

การรับส่งข้อความแบบเท็กซ์โหมด (Text Mode)

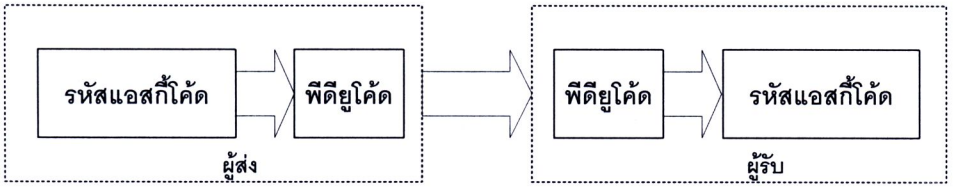


สำนักงานคณะกรรมการวิจัยแห่งชาติ ห้องสมุดการวิจัย

รูปที่ 2.13 การรับส่งข้อความแบบเท็กซ์โหมด

จากรูปที่ 2.13 การรับส่งข้อความแบบเท็กซ์โหมด (Text Mode) ซึ่งผู้ใช้สามารถพิมพ์ตัวอักษรหรือข้อความลงในโทรศัพท์เคลื่อนที่ โดยที่เครื่องจะเก็บตัวอักษรหรือข้อความเป็นรหัสแอสกี้โค้ดและนำรหัสแอสกี้โค้ดที่ได้ส่งให้กับเลขหมายปลายทาง และเลขหมายปลายทางจะต้องแปลงรหัสแอสกี้โค้ดที่ได้มาให้เป็นตัวอักษรเพื่อแสดงผลออกหน้าจอโทรศัพท์เคลื่อนที่

การรับส่งข้อความแบบพีดียูโหมด (PDU Mode)



รูปที่ 2.14 การรับส่งข้อความแบบพีดียูโหมด

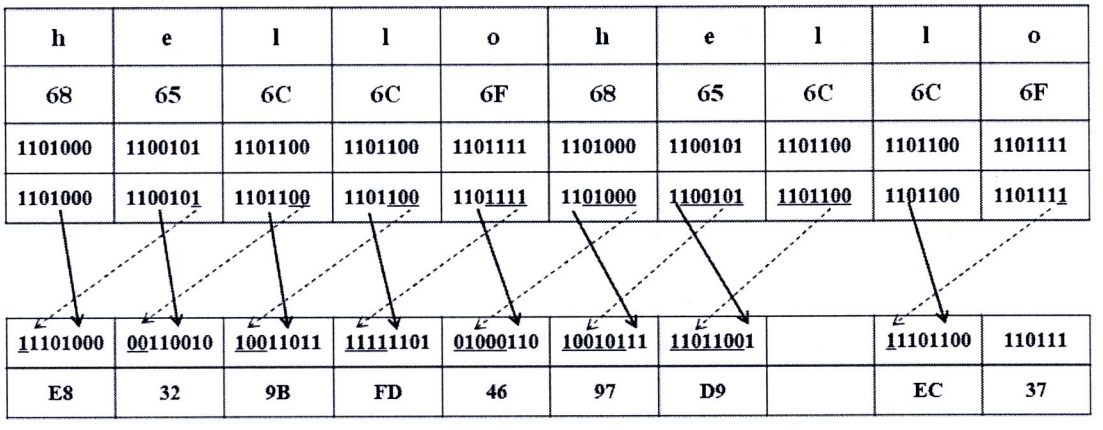
จากรูปที่ 2.14 เป็นบล็อกแสดงการรับส่งข้อความแบบพีดียูโหมด (PDU Mode) กล่าวคือในการเขียนตัวอักษรหรือข้อความจะเหมือนกับเท็กซ์โหมด แต่ในการส่งข้อความจะมีการแปลงจากรหัสแอสกี้โค้ดให้เป็นรหัสพีดียูโค้ดและนำรหัสพีดียูโค้ด ที่ได้ส่งให้กับเลขหมายปลายทาง และเลขหมายปลายทางจะต้องแปลงจากรหัสพีดียูโค้ดให้เป็นรหัสแอสกี้โค้ดต่อไป

2.2.3 การรับข้อความในพีดียูโหมด (PDU mode)

dreamfabric.com (2012) อธิบายไว้ว่า ในข้อความในตัวพีดียู ไม่ได้มีแค่ข้อความอย่างเดียวเท่านั้น แต่ยังมีข้อมูลอื่นอีกด้วย เช่น ข้อมูลเกี่ยวกับผู้ส่ง ,ศูนย์ผู้ให้บริการเอสเอ็มเอส และอื่นๆ ตัวอย่างการส่งข้อความ “hellohello” ซึ่งมีได้ พีดียูโค้ด เป็นดังนี้

07 917283010010F5 040BC87238880900F10000993092516195800A E8329BFD4697D9EC37

ลำดับกลุ่มตัวอักษร 8 บิตนี้ประกอบไปด้วยสามส่วน ดังนี้ส่วนกลุ่มตัวอักษรเริ่มต้น(Initial octet) บ่งชี้ถึงความยาวของ ศูนย์ให้บริการข้อมูล(SMSC information) “07” ส่วนที่สองคือส่วนศูนย์ให้บริการข้อมูล(SMSC information) “917283010010F5” และส่วนสุดท้ายคือส่วนของการส่งเอสเอ็มเอส(SMS_DELIVER part) ซึ่งเป็นไปตามข้อกำหนด GSM 03.40 ข้อความ “hellohello” ประกอบด้วย 10 ตัวอักษร เมื่อแทนด้วย 7 บิตแต่ละตัวอักษรเหล่านี้จะกลายเป็นกลุ่มอักษร 8 บิต สำหรับใช้ในการส่งเอสเอ็มเอส



รูปที่ 2.15 ตัวอย่างการเข้ารหัสพีดียู ของคำว่า hellohello

ขั้นตอนการเข้ารหัสพีดียูมีขั้นตอนดังนี้

1. จะต้องทราบรหัสแอสกีแบบเลขฐาน 16 ของแต่ละตัวอักขระ

2. แปลงจากรหัสแอสกีแบบเลขฐาน 16 เป็นรหัสแอสกี้แบบเลขฐาน 2 แบบตัดบิตซ้ายสุดทิ้งจะได้กลุ่มเลขฐาน 2 แบบ 7 บิต

3. แปลงเป็นรหัสพีดียู โดยการนำเอาบิตทางขาวมือสุดของตัวเลขกลุ่มที่ 2 มาเติมบิตทางซ้ายมือของเลขกลุ่มที่ 1 ให้ครบเป็น 8 บิต จะได้ตัวอักขระตัวที่ 1 และตัวเลขของกลุ่มที่ 2 จะเหลือเป็น 6 บิต ต้องนำเอาบิตทางขาวมือสุดของตัวเลขกลุ่มที่ 3 จำนวน 2 บิตมาเติมบิตทางซ้ายมือของกลุ่มที่ 2 จะได้ตัวอักขระตัวที่ 2 และตัวเลขของกลุ่มที่ 3 จะเหลือเป็น 5 บิต ต้องนำเอาบิตทางขาวมือสุดของตัวเลขกลุ่มที่ 4 จำนวน 3 บิตมาเติมบิตทางซ้ายมือของกลุ่มที่ 3 จะได้ตัวอักขระตัวที่ 3 และตัวเลขของกลุ่มที่ 4 จะเหลือเป็น 4 บิต ต้องนำเอาบิตทางขาวมือสุดของตัวเลขกลุ่มที่ 5 จำนวน 4 บิตมาเติมบิตทางซ้ายมือของกลุ่มที่ 3 จะได้ตัวอักขระตัวที่ 3 จะทำขั้นตอนเดิมต่อไปจนครบทุกอักขระ

4. ทำการแปลงรหัสแอสกี้เลขฐาน 2 ของการทำให้ครบ 8 บิตให้มาอยู่ในรูปเลขฐาน16 จากข้อมูลคำว่า “hellohello” แปลงเป็น รหัสพีดียู เป็น E8 32 9B FD 46 97 D9 EC 37

ตารางที่ 2.5 อธิบายความหมายของโค้ดในแต่ละชุด

Octet(s)	Description
07	ความยาวของ SMSC information(ในที่นี้คือ 7 octets)
91	ชนิดของ Address ของ SMSC (91 หมายถึง internatonal format ของหมายเลขโทรศัพท์)
72 83 01 00 10 F5	Service center number ความยาวของหมายเลขโทรศัพท์เป็นจำนวนคี่ (11) ดังนั้นจึงมีการเพิ่ม F เข้าไปเพื่อให้ครบ octet, หมายเลขโทรศัพท์นี้คือ “+273810005”
04	Octet แรกของ SMS-DELIVER message
0B	ความยาวของที่อยู่ผู้ส่ง
C8	ประเภท ที่อยู่ของหมายเลขผู้ส่ง
72 38 88 90 00 F1	หมายเลขผู้ส่ง เติม F เพื่อให้ครบ octet
00	TP-PID บอก Protocol identifier
00	TP-DSC บอก Data coding
99 30 92 51 61 95 80	TP-SCTS เป็น Time stamp
0A	TP-UDL หรือ user data length ความยาวของ message ในที่นี้ข้อความ “hellohello” มีความยาวเป็น 10 จึงได้เป็น 0A
E8329BFD4697D9EC37	TP-UD เป็นข้อความ “hellohello” เป็น 8-bit octet ที่แสดงเป็น 7-bit data

2.3 เบส64 (Base 64)

วิกิพีเดีย (2012) อธิบายไว้ว่า เบส64 เป็นวิธีเข้ารหัส/ถอดรหัส(encode/decode)ข้อมูล เบส 64 จะใช้สำหรับแปลงข้อมูลที่เป็นไบนารี หรือข้อมูลที่มีตัวอักษรพิเศษ ไปเป็นข้อมูลที่เป็นข้อความ 64 ตัว เป็นที่มาของชื่อ เบส64 คือหนึ่งหลักมีเลขอยู่ 64 ตัว ปรกติก็จะประกอบไปด้วย 0-9 (10 ตัว) A-Z (26 ตัว) a-z (26 ตัว) รวมเป็น 62 ตัวบวกกับสัญลักษณ์ + และ / อีกสองตัว จะสังเกตได้ว่า ถ้าเลือกสัญลักษณ์ดีๆ ผลลัพธ์ของเบส64 จะไปใช้ทำอะไรได้หลายอย่างโดยไม่มีปัญหา เช่น ส่งไปทางจดหมายอิเล็กทรอนิกส์ เป็นต้น

การแปลงข้อมูล(encode)ไปเป็นเบส64 มีหลักอยู่ว่า ข้อมูลปลายทางแต่ละหลัก(ตัว)มีความเป็นไปได้ 64 ตัวอักษร 64 คือ 2 ยกกำลัง 6 ดังนั้นเราต้องการข้อมูลต้นทางแค่ 6 บิต สำหรับข้อมูลปลายทางแต่ละตัว (8 bits) ตัวอย่างการแปลงข้อมูล คำว่า “Man” ดังรูปที่ 2.16 อธิบายได้ว่า

จากคำว่า “Man” รหัสแอสกีประจำของตัวอักษรแตละตัว “M”,”a”,”n” ค่าประจำหลัก 77 97 110 เมื่อแปลงเป็นเลขไบนารี 8 บิตจะได้ 01001101, 01100001, 01101110 จะเห็นด้วากลุ่ม

ตัวอักษรทั้งสามตัวเมื่อรวมกันจะได้ 24 บิต 010011010110000101101110 เมื่อเอาทั้งหมดมาจัดกลุ่มใหม่เป็นกลุ่มละ 6 บิต(การจัดกลุ่ม 6 บิตจะได้ค่าสูงสุด 64 ค่า) จะได้ค่าจากการจัดกลุ่มได้ 4 ค่า 010011 010110 000101 101110 แล้วแปลงกลับมาเป็นเลขฐาน 10 จะได้ 19 22 5 46 แล้วทำการเปรียบเทียบค่าจากตารางเบส64

Text content	**M**								**a**								**n**							
ASCII	77								97								110							
Bit pattern	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Index	19								22								5							
Base64-encoded	**T**								**W**								**F**							

รูปที่ 2.16 การเข้ารหัสแบบเบส64

ในตารางที่ 2.6 เป็นการแสดงให้เห็นว่ารหัสเบส64 ประกอบไปด้วยอะไรบางพร้อมทั้งแสดงเลขฐานสิบประจำหลักของรหัส

ตารางที่ 2.6 รหัสของเบส64

เลขฐาน 10	ตัวอักษร	เลขฐาน 10	ตัวอักษร	เลขฐาน 10	ตัวอักษร	เลขฐาน 10	ตัวอักษร
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

2.4 งานวิจัยที่เกี่ยวข้อง

การพัฒนาโปรแกรมสำหรับรับส่งคิวอาร์โค้ดของรูปภาพผ่านระบบเอสเอ็มเอสบนโทรศัพท์เคลื่อนที่ ผู้ศึกษาได้ค้นคว้างานวิจัยที่เกี่ยวข้องดังนี้

การนำคิวอาร์โค้ดไปใช้งาน กานต์ คงบรรทัด (2551) คณะเกษตรศาสตร์มหาวิทยาลัยเชียงใหม่ ได้นำเทคโนโลยีการตรวจสอบย้อนกลับผ่านโทรศัพท์เคลื่อนที่ด้วยคิวอาร์โค้ดมาใช้กับส้มสายน้ำผึ้งของวิสาหกิจชุมชนผู้ผลิตส้มคุณภาพ อ.แม่อาย จ.เชียงใหม่ เพื่อเพิ่มศักยภาพในการแข่งขันของเกษตรกรและเพิ่มความมั่นใจให้ผู้บริโภค ลักษณะงานเป็นการใช้โทรศัพท์เคลื่อนที่ที่มีกล้องและโปรแกรมอ่านรหัส คิวอาร์โค้ด ทำสามารถรู้รายละเอียดของสินค้าเพิ่มเติม หรือตรวจสอบย้อนกลับได้ทันที ช่วยให้ประกอบการตัดสินใจการซื้อได้ง่ายขึ้น

ทีมนักวิจัยมหาวิทยาลัยเอดิทโคแวน (ECU: Edith Cowan University) ได้ทำการพัฒนาบาร์โค้ด 2 มิติ สามารถบันทึกข้อมูลมัลติมีเดียได้ Alfred Tan and Douglas Chai (2552) ทั้งยังใช้งานดาวน์โหลดไฟล์เหล่านั้นได้โดยที่ไม่ต้องเชื่อมต่ออินเทอร์เน็ต โดยใช้ชื่อว่า เอ็มเอ็มซีซี (MMCC: Mobile Multi-Colour Composite 2D-Barcode)

Mohammad Shirali-Shahrezaและ Sajad Shirali-Shahreza Sharif University of Technology เตหะรานประเทศอิหร่าน ได้นำเสนอวิธีการสำหรับการแก้ปัญหาที่เกิดขึ้นจากการส่งภาพโดยโทรศัพท์มือถือเมื่อไม่มีบริการส่ง เอ็มเอ็มเอส โดยใช้คุณสมบัติของเอสเอ็มเอส เข้ามาใช้ในการส่งภาพโดยภาพที่ใช้ในการส่งจะเป็นภาพที่มีคุณภาพต่ำมีเพียงสองสี(ขาวดำ)และมีขนาดเล็กโดยวิธีการเข้ารหัสรูปภาพในแบบเบส64 (Base 64) เพื่อนำเอาข้อความของการเข้ารหัสมาทำการส่งข้อความผ่านระบบเอสเอ็มเอส

จากงานวิจัยที่เกี่ยวข้องจะเห็นได้ว่ามีการนำบาร์โค้ด 2 มิติมาใช้ในด้านการเก็บข้อมูลไม่ว่าจะเป็นบาร์โค้ดแบบคิวอาร์โค้ด หรือ แบบเอ็มเอ็มซีซีที่ถูกพัฒนาต่อยอดมาจากคิวอาร์โค้ดเพื่อที่จะทำการเก็บข้อมูลได้เยอะกว่าและผู้ศึกษาได้เจองานวิจัยที่เกี่ยวกับการส่งภาพผ่านระบบเอสเอ็มเอสซึ่งใช้การเข้ารหัสแบบเบส64 เพื่อให้สอดคล้องกับความต้องการของระบบการส่งข้อความเอสเอ็มเอสให้สามารถส่งได้และได้ทำการศึกษาเพื่อเป็นแนวทางของการทำงานต่อ

สรุป

ในบทนี้ได้กล่าวถึงทฤษฎีที่เกี่ยวข้องกับการรับ-ส่งคิวอาร์โค้ดของรูปภาพผ่านระบบเอสเอ็มเอสที่ประกอบไปด้วยข้อมูลของคิวอาร์โค้ด ไม่ว่าจะเป็นขนาดของความจุข้อมูลในคิวอาร์โค้ด โครงสร้างของคิวอาร์โค้ด ระดับความผิดพลาดของคิวอาร์โค้ด ตารางแสดงรายละเอียดของคิวอาร์

โค้ดในแต่ละเวอร์ชัน และการถอดรหัสในคิวอาร์โค้ด และในส่วนของการส่งเอสเอ็มเอส ไม่ว่าจะเป็นการทำงานของระบบการรับส่งข้อความ โหมดของการรับส่งของข้อมูลเอสเอ็มเอส การรับข้อความในพีดียูโหมด และส่วนสุดท้ายเป็นการสรุปแนวทางงานวิจัยที่เกี่ยวข้อง ซึ่งความรู้ที่ได้จากทฤษฎีและงานวิจัยเหล่านี้สามารถนำไปใช้เป็นความรู้สำหรับนำไปพัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ใช้ในการรับส่งคิวอาร์โค้ดของรูปภาพผ่านระบบเอสเอ็มเอส ในบทต่อไป