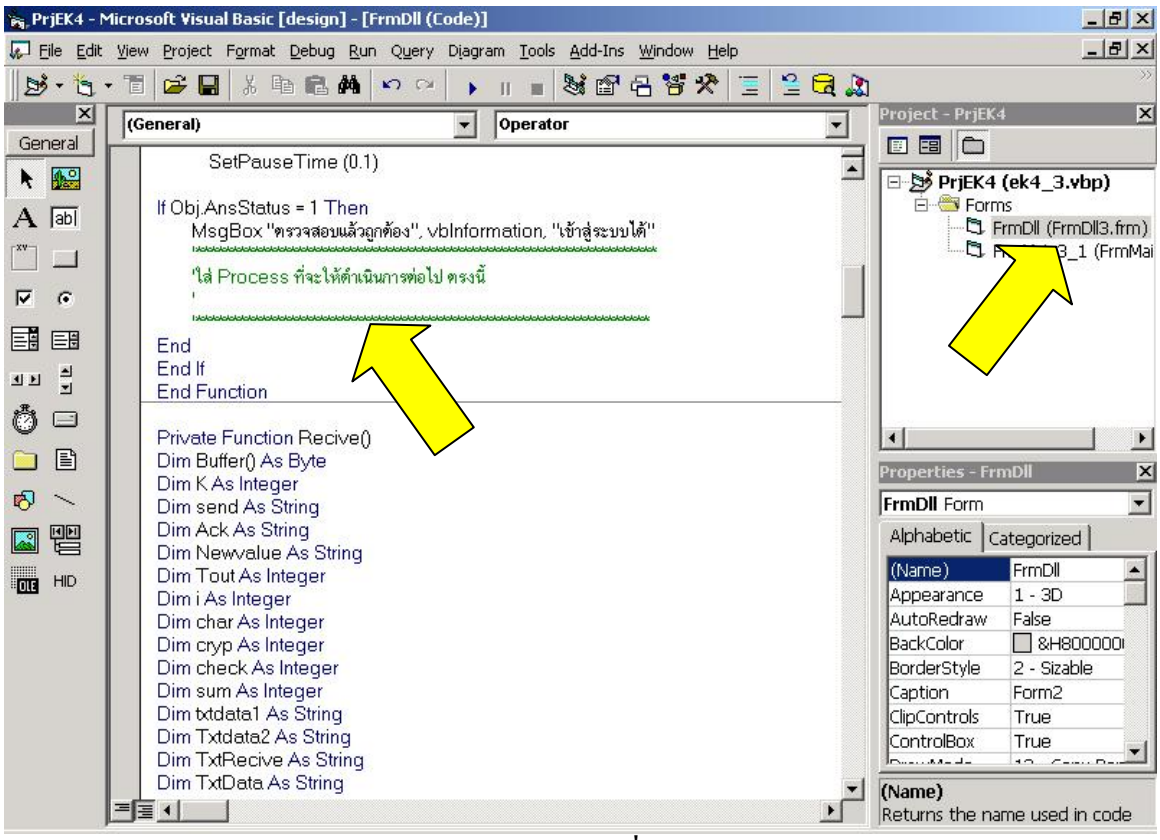
ก1. การแทรกโค้ดโปรแกรม

เมื่อโปรแกรมเมอร์นำกุญแจอิเล็กทรอนิกส์มาประยุกต์ใช้กับโปรแกรมต่าง ๆ สามารถทำได้โดยการแทรกโค้ดลงไปในฟอร์มดังแสดงตามภาพประกอบ ก-1

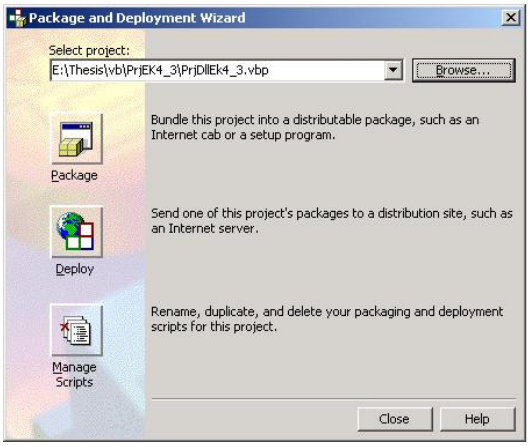


ภาพประกอบ ก-1 การแทรก Code โปรแกรมเพื่อประยุกต์ใช้งาน กุญแจอิเล็กทรอนิกส์

จากภาพประกอบ ก-1 โปรแกรมเมอร์สามารถนำกุญแจอิเล็กทรอนิกส์ไปประยุกต์ใช้ได้โดยการแทรกโค้ดเข้าไปในโปรแกรมที่ต้องการจำกัดสิทธิ์ ซึ่งผู้วิจัยได้จัดเตรียมไว้ให้ โดยที่โปรแกรมเมอร์ต้องแทรกเข้าไปในฟอร์ม ชื่อ FrmDll (FrmDll3.frm) ซึ่งหากต้องการให้กุญแจอิเล็กทรอนิกส์ไปควบคุมโปรแกรมใด สามารถแทรกเข้าไปในส่วนนี้ได้โดยตรง

ก2. การจัดเตรียมโปรแกรมเพื่อนำไปติดตั้งที่เครื่องลูกข่าย

หลังจากที่โปรแกรมเมอร์แทรกโค้ด ลงไปในโปรแกรมที่ต้องการจำกัดสิทธิ์เรียบร้อยแล้ว เมื่อต้องการนำโปรแกรมไปติดตั้งที่เครื่องลูกข่าย จำเป็นต้องจัดเตรียมโปรแกรมดังกล่าวก่อนนำไปติดตั้งจริง ซึ่งวิธีการจัดเตรียมสามารถทำโดย Start > Microsoft Visual Studio 6.0 > Microsoft Visual Studio 6.0 Tool > Package & Development wizard จากนั้นดำเนินการดังนี้

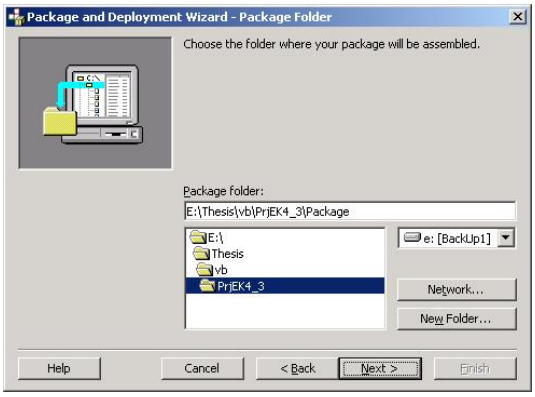


ภาพประกอบ ก- 2 Package wizard

- เลือก Project ที่จะทำการเตรียมติดตั้ง
- กดปุ่ม Package เพื่อดำเนินการต่อ

ภาพประกอบ ก-3 Package Type

- เลือกชนิดการจัดเตรียมเป็นแบบ Standard
- กดปุ่ม Next เพื่อดำเนินการต่อ

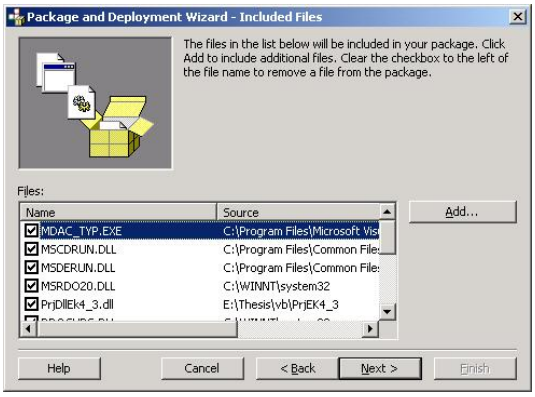


ภาพประกอบ ก-4 Package Folder

- เลือกตำแหน่งที่จัดเก็บเมื่อจัดเตรียมเรียบร้อยแล้ว
- กดปุ่ม Next เพื่อดำเนินการต่อ

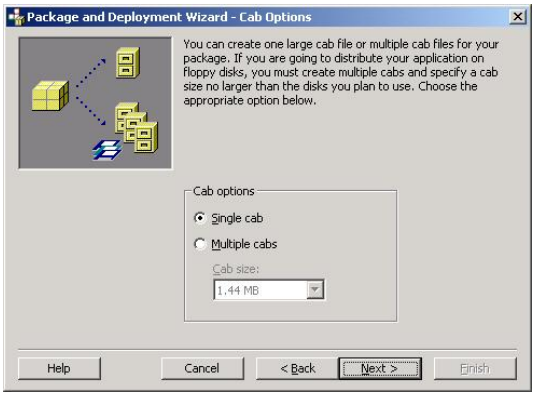
ภาพประกอบ ก-5 Create Folder

- กด Yes เพื่อสร้าง Folder ใหม่ ซึ่งใช้สำหรับเก็บชุด Package ไว้ภายใน



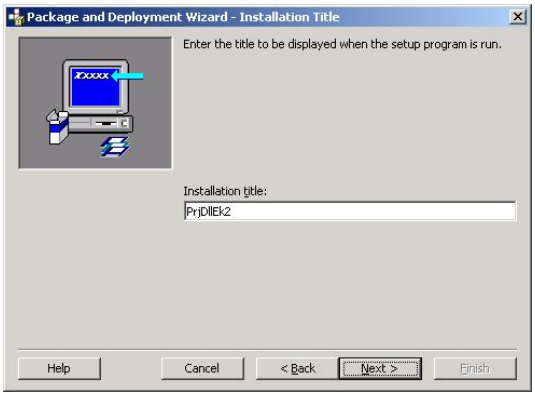
ภาพประกอบ ก-6 Include Files

- ระบบแจ้งให้ทราบถึงไฟล์ที่จะรวมไปกับการจัดเตรียมครั้งนี้เช่น PrjDll4_3.dll ซึ่งเป็นไฟล์ DLL ที่ใช้ในการตรวจสอบ กุญแจอิเล็กทรอนิกส์ และไฟล์อื่น ๆ ที่ระบบต้องการ
- กด Next เพื่อดำเนินการต่อ



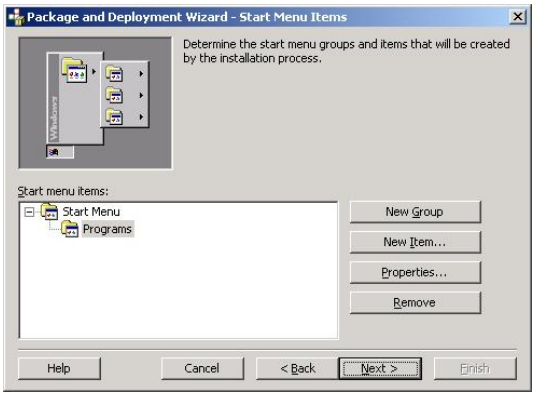
ภาพประกอบ ก- 7 Cab Options

- เลือก Single cab เพื่อบรรจุชุดจัดเตรียมลงในฮาร์ดดิสก์
- กด Next เพื่อดำเนินการต่อ



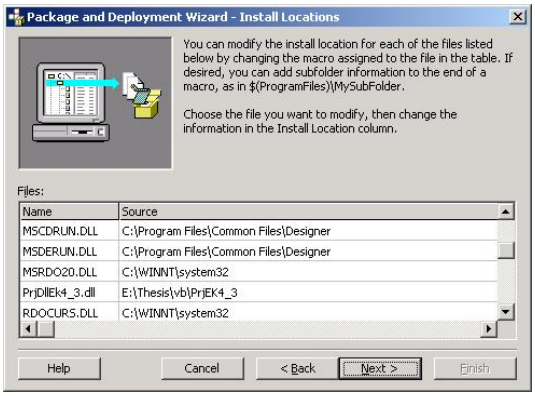
ภาพประกอบ ก-8 Installation Title

- กำหนดชื่อในการติดตั้งโดยชื่อนี้จะปรากฏเมื่อนำชุดจัดเตรียมนี้ไปติดตั้งที่เครื่องลูกข่าย
- กด Next เพื่อดำเนินการต่อ



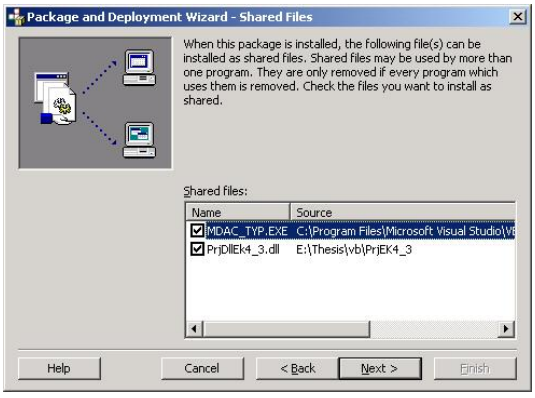
ภาพประกอบ ก-9 Start Menu Items

- เลือกกลุ่มที่จะให้ชุดจัดเตรียมนี้ติดตั้งลงไปในเครื่องลูกข่าย
- กด Next เพื่อดำเนินการต่อ



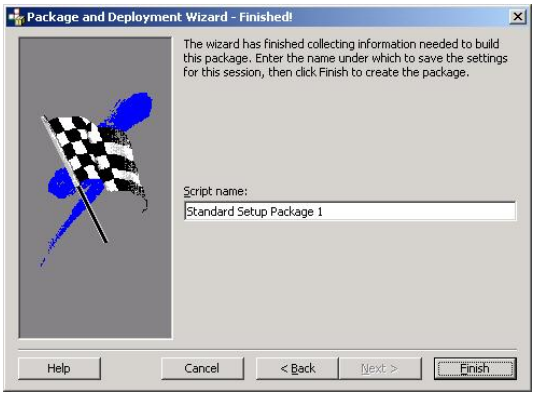
ภาพประกอบ ก-10 Install Locations

- ระบบแจ้งให้ทราบถึง install locations ที่จะนำไฟล์ใดเข้าร่วมกับชุดจัดเตรียมโดยสามารถปรับแต่งได้อีกครั้งก่อนจะจัดเตรียม จริง
- กด Next เพื่อดำเนินการต่อ



ภาพประกอบ ก-11 Shared Files

- ระบบแจ้งให้ทราบว่าจะมีไฟล์ถูกใช้งานร่วมกัน (Share file) หรือไฟล์ถูกใช้งานมากกว่า 1 แอปปลิเคชั่น
- กด Next เพื่อดำเนินการต่อ



ภาพประกอบ ก-12 Finished

- ระบบแจ้งให้ทราบว่าขณะนี้ ได้ดำเนินการ ชัดเตรียมเรียบร้อยแล้ว
- กด Next เพื่อตรวจสอบรายงานสรุป

ภาพประกอบ ก-13 Packaging Report

- รายงานผลการจัดเตรียมทั้งหมด

ข1.รายละเอียดทางเทคนิคของบอร์ด

รุ่น U-Board

ภาพประกอบ ข-1 บอร์ด U Board

คุณสมบัติทางเทคนิค

- PIC16C745
- 6 MHz clock frequency
- รองรับ USB 1.0/1.1
- ใช้ไฟเลี้ยงได้จากพอร์ต USB และจากภายนอก
- มี Connector ขยายระบบแบ่งเป็น PORTOUT,PORTIN และ SHIFTOUT
- 6*12 cm.

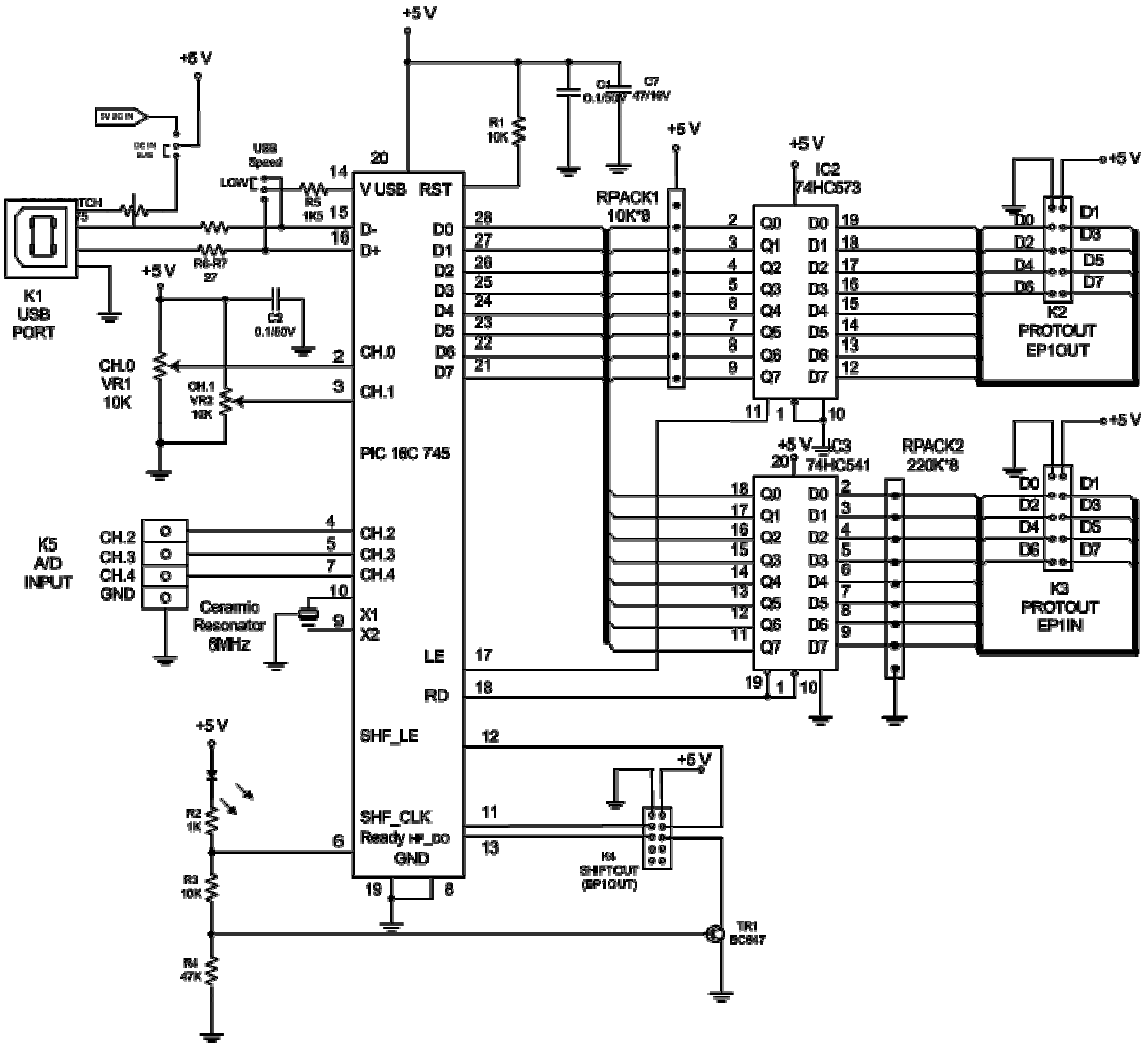
รุ่น JX-876

ภาพประกอบ ข- 2 บอร์ด JX-876

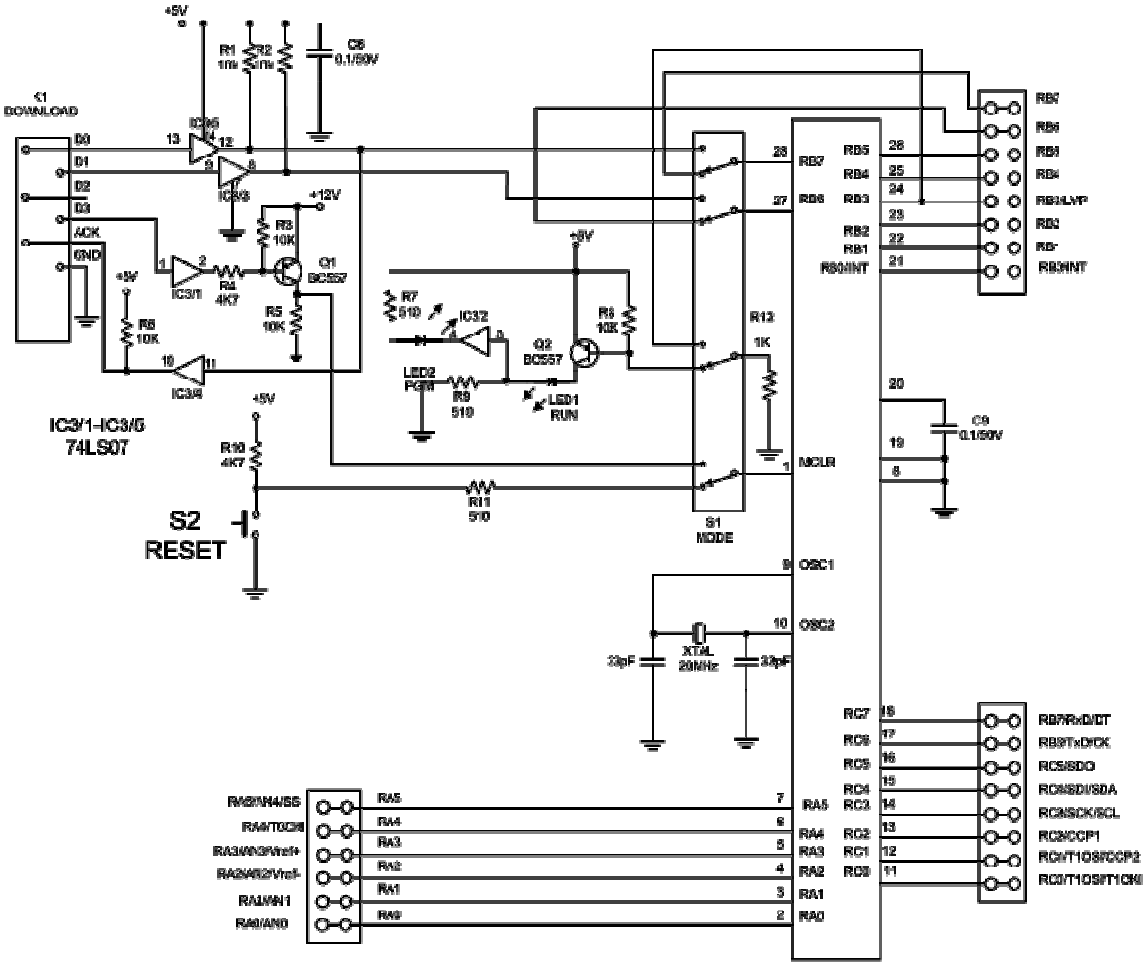
คุณสมบัติทางเทคนิค

- PIC18F242
- 20 MHz clock frequency
- full 22 I/O
- Download code to chip's memory directly
- Select RUN or PROGRAM mode by on switch with LED mode status
- 6*12 cm.

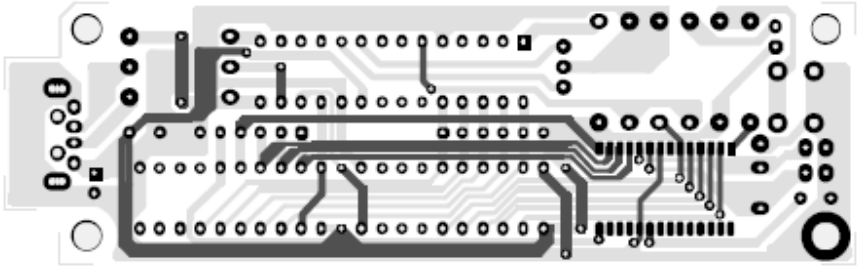
ข2. รายละเอียดวงจรของบอร์ด



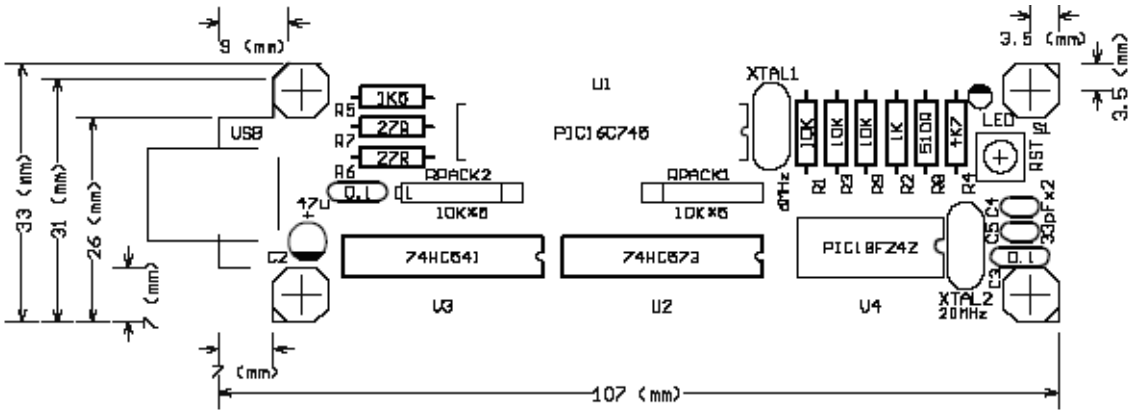
ภาพประกอบ ข- 3 วงจร U Board



ภาพประกอบ ข- 4 วงจร JX-876 Board



ภาพประกอบ ข- 5 ลายวงจรกุญแจอิเล็กทรอนิกส์



ภาพประกอบ ข-6 การวางอุปกรณ์ของกุญแจอิเล็กทรอนิกส์

ค1.ตารางการคำนวณเพื่อหาประสิทธิภาพการลดทอนแบบ Montgomery

เมื่อ M=26, e=7, M^e = 8.03E+09

n	r	n'	Function MomExp		Function MonPro	
			step2. $\bar{M}$	step3. $\bar{C}$	step1. t	step2. tn'
35	64	117	19	29	841	98397
65	128	191	13	63	3969	758079
77	128	123	17	51	2601	319923
95	128	97	3	33	1089	105633
119	128	183	115	9	1035	189405
143	256	401	78	113	12769	5120369
145	256	399	131	111	14541	5801859
185	256	375	181	71	12851	4819125
209	256	207	177	47	8319	1722033
221	256	395	26	35	1225	483875
319	512	833	233	193	44969	37459177
323	512	661	69	189	35721	23611581
407	512	473	288	105	30240	14303520
493	512	539	1	19	361	194579
629	1024	1571	206	395	156025	245115275
851	1024	805	243	173	42039	33841395
1147	2048	3405	486	901	811801	2764182405
8827	16384	21325	2288	7557	57108249	1.21783E+12

เมื่อ M=37, e=7, M^e = 9.49E+09

n	r	n'	Function MomExp		Function MonPro	
			step2. $\bar{M}$	step3. $\bar{C}$	step1. t	step2. tn'
35	64	117	23	29	841	98397
65	128	191	56	63	3969	758079
77	128	123	39	51	2601	319923
95	128	97	81	33	2673	259281
119	128	183	95	9	855	156465
143	256	401	34	113	12769	5120369
145	256	399	47	111	12321	4916079
185	256	375	37	71	5041	1890375
209	256	207	67	47	3149	651843
221	256	395	190	35	6650	2626750
319	512	833	123	193	37249	31028417
323	512	661	210	189	39690	26235090
407	512	473	222	105	23310	11025630
493	512	539	210	19	3990	2150610
629	1024	1571	148	395	156025	245115275
851	1024	805	444	173	76812	61833660
1147	2048	3405	74	901	811801	2764182405
8827	16384	21325	5972	7557	57108249	1.21783E+12

เมื่อ M=50, e=7, M^e = 7.81E+11

n	r	n'	Function MomExp		Function MonPro	
			step2. $\bar{M}$	step3. $\bar{C}$	step1. t	step2. tn'
35	64	117	15	29	841	98397
65	128	191	30	63	3969	758079
77	128	123	9	51	2601	319923
95	128	97	35	33	1155	112035
119	128	183	93	9	837	153171
143	256	401	73	113	12769	5120369
145	256	399	40	111	12321	4916079
185	256	375	35	71	5041	1890375
209	256	207	51	47	2397	496179
221	256	395	203	35	7105	2806475
319	512	833	80	193	37249	31028417
323	512	661	83	189	35721	23611581
407	512	473	366	105	38430	18177390
493	512	539	457	19	8683	4680137
629	1024	1571	251	395	156025	245115275
851	1024	805	140	173	29929	24092845
1147	2048	3405	317	901	811801	2764182405
8827	16384	21325	7116	7557	57108249	1.21783E+12

เมื่อ M=73, e=7, M^e = 1.1E+13

n	r	n'	Function MomExp		Function MonPro	
			step2. $\bar{M}$	step3. $\bar{C}$	step1. t	step2. tn'
35	64	117	17	29	841	98397
65	128	191	49	63	3969	758079
77	128	123	27	51	2601	319923
95	128	97	34	33	1122	108834
119	128	183	62	9	558	102114
143	256	401	98	113	12769	5120369
145	256	399	128	111	14208	5668992
185	256	375	3	71	5041	1890375
209	256	207	87	47	4089	846423
221	256	395	124	35	4340	1714300
319	512	833	53	193	37249	31028417
323	512	661	231	189	43659	28858599
407	512	473	339	105	35595	16836435
493	512	539	401	19	7619	4106641
629	1024	1571	530	395	209350	328888850
851	1024	805	715	173	123695	99574475
1147	2048	3405	394	901	811801	2764182405
8827	16384	21325	4387	7557	57108249	1.21783E+12

เมื่อ M=100, e=7, M^e = 1E+14

n	r	n'	Function MomExp		Function MonPro	
			step2. $\bar{M}$	step3. $\bar{C}$	step1. t	step2. tn'
35	64	117	30	29	870	101790
65	128	191	60	63	3969	758079
77	128	123	18	51	2601	319923
95	128	97	70	33	2310	224070
119	128	183	67	9	603	110349
143	256	401	3	113	12769	5120369
145	256	399	80	111	12321	4916079
185	256	375	70	71	5041	1890375
209	256	207	102	47	4794	992358
221	256	395	185	35	6475	2557625
319	512	833	160	193	37249	31028417
323	512	661	166	189	35721	23611581
407	512	473	325	105	34125	16141125
493	512	539	421	19	7999	4311461
629	1024	1571	502	395	198290	311513590
851	1024	805	280	173	48440	38994200
1147	2048	3405	634	901	811801	2764182405
8827	16384	21325	5405	7557	57108249	1.21783E+12

เมื่อ M=130, e=7, M^e = 6.27E+14

n	r	n'	Function MomExp		Function MonPro	
			step2. $\bar{M}$	step3 $\bar{C}$	step1. t	step2. tn'
35	64	117	25	29	841	98397
65	128	191	0	63	3969	758079
77	128	123	8	51	2601	319923
95	128	97	15	33	1089	105633
119	128	183	99	9	891	163053
143	256	401	104	113	12769	5120369
145	256	399	75	111	12321	4916079
185	256	375	165	71	11715	4393125
209	256	207	49	47	2303	476721
221	256	395	130	35	4550	1797250
319	512	833	208	193	40144	33439952
323	512	661	22	189	35721	23611581
407	512	473	219	105	22995	10876635
493	512	539	5	19	361	194579
629	1024	1571	401	395	158395	248838545
851	1024	805	364	173	62972	50692460
1147	2048	3405	136	901	811801	2764182405
8827	16384	21325	2613	7557	57108249	1.21783E+12

เมื่อ M=175, e=7, M^e = 5.03E+15

n	r	n'	Function MomExp		Function MonPro	
			step2. $\bar{M}$	step3. $\bar{C}$	step1. t	step2. tn'
35	64	117	0	29	841	98397
65	128	191	40	63	3969	758079
77	128	123	70	51	3570	439110
95	128	97	75	33	2475	240075
119	128	183	28	9	252	46116
143	256	401	41	113	12769	5120369
145	256	399	140	111	15540	6200460
185	256	375	30	71	5041	1890375
209	256	207	74	47	3478	719946
221	256	395	158	35	5530	2184350
319	512	833	280	193	54040	45015320
323	512	661	129	189	35721	23611581
407	512	473	60	105	11025	5214825
493	512	539	367	19	6973	3758447
629	1024	1571	564	395	222780	349987380
851	1024	805	490	173	84770	68239850
1147	2048	3405	536	901	811801	2764182405
8827	16384	21325	7252	7557	57108249	1.21783E+12

เมื่อ M=255, e=7, M^e = 7.01E+16

n	r	n'	Function MomExp		Function MonPro	
			step2. $\bar{M}$	step3. $\bar{C}$	step1. t	step2. tn'
35	64	117	10	29	841	98397
65	128	191	10	63	3969	758079
77	128	123	69	51	3519	432837
95	128	97	55	33	1815	176055
119	128	183	34	9	306	55998
143	256	401	72	113	12769	5120369
145	256	399	30	111	12321	4916079
185	256	375	160	71	11360	4260000
209	256	207	72	47	3384	700488
221	256	395	85	35	2975	1175125
319	512	833	89	193	37249	31028417
323	512	661	68	189	35721	23611581
407	512	473	320	105	33600	15892800
493	512	539	408	19	7752	4178328
629	1024	1571	85	395	156025	245115275
851	1024	805	714	173	123522	99435210
1147	2048	3405	355	901	811801	2764182405
8827	16384	21325	2749	7557	57108249	1.21783E+12

ค2.ความเร็วในการประมวลผลการเข้ารหัสแบบ RSA และ RSA with Montgomery Reduction

เมื่อ n=119, ขนาดโค้ด=1 Kbyte และขนาดไฟล์ Execute = 26 Kbyte

RSA

m	e	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	เฉลี่ย
19	3	0.0115	0.0122	0.0122	0.0122	0.0121	0.0123	0.0122	0.0121
19	5	0.0121	0.0121	0.0123	0.0121	0.0121	0.0122	0.0121	0.0121
19	7	0.0123	0.0113	0.0113	0.0114	0.0124	0.0126	0.0123	0.0119
21	3	0.0124	0.0122	0.0125	0.0124	0.0122	0.0122	0.0114	0.0122
21	5	0.0123	0.0125	0.0122	0.0123	0.0123	0.0122	0.0123	0.0123
21	7	0.0122	0.0121	0.0126	0.0116	0.0122	0.0115	0.0122	0.0121
23	3	0.0122	0.0122	0.0121	0.0123	0.0126	0.0126	0.0121	0.0123
23	5	0.0122	0.0122	0.0122	0.0122	0.0123	0.0122	0.0123	0.0122
23	7	0.0122	0.0123	0.0125	0.0124	0.0122	0.0122	0.0113	0.0122
37	3	0.0113	0.0112	0.0122	0.0126	0.0125	0.0126	0.0121	0.0121
37	5	0.0125	0.0122	0.0126	0.0125	0.0122	0.0123	0.0125	0.0124
73	3	0.0125	0.0121	0.0122	0.0121	0.0122	0.0123	0.0122	0.0122
73	7	0.0128	0.0128	0.0132	0.0131	0.0128	0.0134	0.0132	0.0130

เมื่อ n=119, ขนาดโค้ด = 4 Kbyte และขนาดไฟล์ Execute = 14 Kbyte

RSA with Montgomery Reduction

m	e	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 3	ครั้งที่ 4	ครั้งที่ 5	ครั้งที่ 6	ครั้งที่ 7	เฉลี่ย
19	3	0.0071	0.0071	0.0072	0.0077	0.0071	0.0072	0.0071	0.0072
19	5	0.0077	0.0076	0.0078	0.0079	0.0076	0.0076	0.0076	0.0077
19	7	0.0079	0.0077	0.0079	0.0077	0.0078	0.0077	0.0081	0.0078
21	3	0.0061	0.0062	0.0061	0.0062	0.0062	0.0061	0.0061	0.0061
21	5	0.0071	0.0070	0.0070	0.0071	0.0072	0.0071	0.0071	0.0071
21	7	0.0072	0.0072	0.0071	0.0074	0.0073	0.0072	0.0073	0.0072
23	3	0.0063	0.0062	0.0067	0.0072	0.0072	0.0072	0.0075	0.0069
23	5	0.0081	0.0081	0.0078	0.0077	0.0072	0.0078	0.0080	0.0078
23	7	0.0077	0.0079	0.0078	0.0076	0.0080	0.0079	0.0076	0.0078
37	3	0.0063	0.0074	0.0063	0.0066	0.0063	0.0072	0.0073	0.0068
37	5	0.0076	0.0080	0.0079	0.0079	0.0077	0.0076	0.0071	0.0077
73	3	0.0066	0.0072	0.0064	0.0064	0.0067	0.0072	0.0075	0.0069
73	7	0.0079	0.0095	0.0079	0.0077	0.0080	0.0077	0.0076	0.0080

การพัฒนา RSA ในกุญแจอิเล็กทรอนิกส์

RSA Implementation of Electronic Key

ชลากร ครุพงศ์สิริ* ผศ.ดร.เกริกชัย ทองหนู ผศ.สาวิตร์ ตัณฑนุช
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ อ.หาดใหญ่ จ.สงขลา 90112
E-mail: chalakorn.s@psu.ac.th

Chalakorn Karupongsiri* Asst.Prof. Dr. Krerkchai Tongnoo Asst.Prof. Sawit Tanthanuch
Department of Electrical Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Songkhla 90112
E-mail: chalakorn.s@psu.ac.th

บทคัดย่อ

บทความนี้นำเสนอการประยุกต์ใช้ Microcontroller เป็นกุญแจอิเล็กทรอนิกส์ ที่มีการเข้ารหัสลับแบบ RSA เพื่อกำหนดสิทธิ์การใช้ Application Software ผ่านทาง USB Port โดยใช้ Montgomery reduction algorithm แบบทวินามเพื่อลดความซับซ้อนในการประมวลผลรหัสลับแบบ RSA ผลการวิจัยสามารถสร้างการประมวลผลรหัสลับแบบ RSA ใน Microcontroller เบอร์ PIC18F242 และใช้ Microcontroller เบอร์ PIC16C745 เชื่อมต่อกับเครื่องคอมพิวเตอร์ส่วนบุคคลผ่าน USB Port ผลการทดสอบสามารถเข้ารหัสลับแบบ RSA บน Microcontroller ได้ทำให้สามารถกำหนดสิทธิ์การใช้ Application Software ได้โดยการพัฒนาด้วยโปรแกรม Visual Basic 6.0

คำหลัก Electronic Key, RSA Cryptography, Montgomery reduction algorithm, Microcontroller

Abstract

This paper presented to implement Microcontroller is about Electronic Key with cryptography RSA used for authentication application software via USB port. The Electronic Key uses Montgomery reduction algorithm in order to decrease complexity in the process of cryptography RSA. The result of this research is that process of cryptography RSA can be implemented with the microcontroller number PIC18F242. And the microcontroller number PIC16C745 is used to interface with personal computer via USB port. The results of the test are to cryptography in microcontroller and authentication application software by implement with Visual Basic 6.0

Keywords: Electronic Key, RSA Cryptography, Montgomery reduction algorithm, Microcontroller

1. บทนำ

การปกป้องความปลอดภัยของ Application Software ซึ่งเป็นโปรแกรมประยุกต์ใช้ทั่วไปเช่น โปรแกรมระบบบัญชี เป็นสิ่งสำคัญในปัจจุบัน เนื่องจากมูลค่าของ Application Software ได้รับความสนใจจากผู้ใช้งานทั่วไป รวมถึงผู้บุกรุกที่พยายามเจาะระบบความเสียหายในทางธุรกิจและองค์กรที่เกิดจากการคุกคามของผู้ไม่ประสงค์ดีถูกให้ความสำคัญมากขึ้นตามลำดับ แต่หาก Application Software มีระบบควบคุมความปลอดภัยที่ดีจะช่วยลดโอกาสเสี่ยงต่อการถูกคุกคามได้

การพิสูจน์ตัวตน เป็นวิธีป้องกันขั้นพื้นฐานที่สำคัญของการรักษาความปลอดภัยในการใช้งาน Application Software โดยทั่วไปการแสดงสิทธิ์โดยใช้ Username และ Password เพื่อกำหนดสิทธิ์การใช้ Software เหล่านั้น ในบางระบบที่ต้องการความปลอดภัยสูงๆ เช่น ระบบการเงิน การพิสูจน์ตัวตนโดยการนำบัญชีผู้ใช้ซึ่งประกอบด้วย Username และ Password ยังไม่เพียงพอต่อความปลอดภัยเนื่องจาก Username และ Password ผู้บุกรุกสามารถทราบได้ และในบางครั้งผู้ใช้ใช้อย่างไม่ระมัดระวัง อาจมีผู้บุกรุกดักจับ Password และนำไปใช้งานได้ ความพยายามของผู้บุกรุกเพื่อเข้าถึงระบบเกิดขึ้นพร้อมๆ กับการใช้งาน Application Software เหล่านั้น ผู้บุกรุกมีความพยายามเพิ่มมากขึ้น เนื่องจากมูลค่าของ Application Software ที่ถูกจำกัดสิทธิ์มีมูลค่าเพิ่มขึ้น ในอดีตการป้องกันผู้บุกรุกผู้ผลิต Application Software ป้องกันการบุกรุกด้วยการสร้าง Hardware โดยเรียกว่า Hard lock[1] ซึ่ง

ลักษณะของ Hard lock จะคอยเฝ้าระวังการใช้งาน Application Software เพื่อจำกัดสิทธิ์การใช้ Application Software แต่การใช้ Hard lock ในลักษณะดังกล่าวมีจุดอ่อนหลายประการเช่น ไม่สะดวกในการใช้งาน เนื่องจากต้องต่อกับ Parallel Port และระบบไม่ซับซ้อนผู้บุกรุกสามารถที่จะทำซ้ำ Hard lock ขึ้นมาใหม่ได้ เมื่อเทคโนโลยีทางด้าน Hardware พัฒนา USB Port ขึ้นเป็น Port มาตรฐานของเครื่องคอมพิวเตอร์ส่วนบุคคล Hard lock จึงถูกพัฒนาขึ้นจากบริษัทชั้นนำซึ่งการสร้าง Hard lock ที่มี Interface ผ่าน USB Port มีจุดเด่นคือผู้ใช้สามารถใช้งานได้สะดวก รวมถึงระบบมีความซับซ้อนเพิ่มมากขึ้น เนื่องจากสามารถเข้ารหัสบนตัว Hard lock ได้ และมีระบบป้องกันการทำซ้ำ ทำให้ป้องกันระบบจากผู้บุกรุกได้เป็นอย่างดี แต่มีจุดด้อยคือต้องนำเข้าเทคโนโลยีจากต่างประเทศ ดังนั้นผู้วิจัยจึงสร้าง Electronic Key ขึ้นมาใช้เองเพื่อลดการนำเข้าเทคโนโลยี โดยยังคงจุดเด่นของ Hard lock เอาไว้

2. ทฤษฎีและหลักการทำงาน

ภายในกุญแจอิเล็กทรอนิกส์จะประกอบด้วย Microcontroller 2 ตัว โดยตัวแรกจะทำหน้าที่ interface กับ USB Port ของเครื่องคอมพิวเตอร์และ Microcontroller อีกตัวหนึ่งจะทำหน้าที่เข้ารหัสแบบ RSA โดยใช้ทฤษฎีการลดทอนของ Montgomery reduction algorithm ช่วยลดทอนสมการ การเข้ารหัสแบบ RSA

2.1 ลักษณะทั่วไปของ USB Port [2]

Port ต่างๆ ในเครื่องคอมพิวเตอร์ส่วนบุคคล ถูกออกแบบมาเพื่อใช้งานเฉพาะอย่าง ทำให้อุปกรณ์แต่ละตัวต้องเลือกใช้ Port ต่างชนิดกันไป เช่น Serial Port ใช้ต่อกับ Modem และ Mouse Parallel Port ใช้ต่อกับเครื่องพิมพ์ และ Port อื่นๆ ทำให้ผู้ใช้งานทั่วไปเกิดความยุ่งยาก และไม่แน่ใจถึงความเสียหายที่อาจเกิดขึ้นหากต่ออุปกรณ์กับ Port ผิดชนิด นอกจากนี้ในการออกแบบคอมพิวเตอร์หากมี Port มากๆ โอกาสจะเกิดปัญหาการแย่งกันใช้ Interrupt Request ซึ่งเป็นส่วนหนึ่งที่จำกัดการติดต่อกับอุปกรณ์ภายนอก ของคอมพิวเตอร์ ทำให้เกิดแนวคิดที่จะกำหนดมาตรฐานเพื่อสร้างเป็น Port ที่ทำให้การติดต่อกับอุปกรณ์ภายนอกอยู่ในรูปแบบเดียวกัน ง่ายสำหรับผู้ใช้ทั่วไป และไม่มีข้อจำกัดในการใช้ Interrupt Request นั้นคือ Universal Serial Bus (USB) มีลักษณะเด่นดังนี้

- ใช้ Connector เพียงชนิดเดียว ต่อกับอุปกรณ์ได้ทุกชนิดเช่น Mouse, Keyboard, Speaker, Hard disk, Printer เป็นต้น
- สามารถรองรับอุปกรณ์ได้มากที่สุด 127 อุปกรณ์
- ไม่เกิดการแย่งกันใช้ Interrupt Request เนื่องจากอุปกรณ์ทุกตัวจะมีหมายเลขประจำตัว (Product ID) ทำให้แม้ว่า USB จะเป็นการสื่อสารแบบอนุกรม แต่ข้อมูลจะถึงปลายทางได้โดยเลขประจำตัวของอุปกรณ์
- ทำงานอัตโนมัติเมื่อต่ออุปกรณ์เข้ากับเครื่องคอมพิวเตอร์
- มีแรงดันจ่ายให้อุปกรณ์ 5 V เพื่ออำนวยความสะดวกกับอุปกรณ์แบบพกพาเช่น Handy Drive, Hard disk USB เป็นต้น
- มีกระแสจ่ายให้อุปกรณ์ 500 mA
- ความเร็วสูงสุดที่ 480 Mbps ที่ความเร็ว High Speed ตามมาตรฐาน USB 2.0

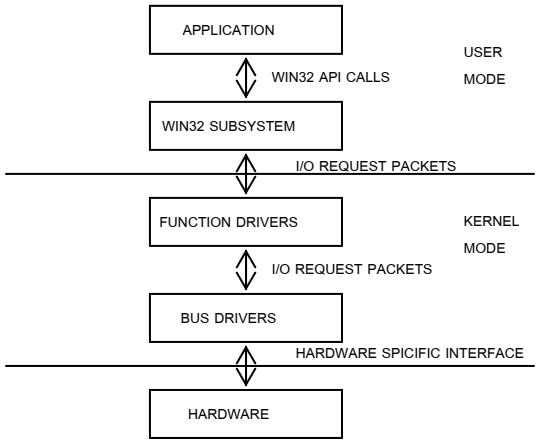
2.2 การเขียนโปรแกรมติดต่อกับ USB Port

หลักการทำงานของเครื่องคอมพิวเตอร์เมื่อต้องการติดต่อกับอุปกรณ์ USB ในที่นี้จะกล่าวถึงเฉพาะระบบปฏิบัติการ Windows 98 SE ขึ้นไป เนื่องจาก Windows ที่ต่ำกว่านี้ไม่สามารถใช้อุปกรณ์ USB ได้หรือใช้ได้แต่ Microsoft ไม่แนะนำ ส่วนระบบปฏิบัติการอื่นๆจะไม่ขอกล่าวถึง

เมื่อ Software application ต้องการติดต่อกับ Hardware ซึ่งเป็น USB Device จะต้องผ่านขั้นตอนกระบวนการที่ Microsoft กำหนดไว้ดังนี้

Application ซึ่งจะอยู่ด้านบนสุด จะต้องติดต่อผ่าน DLL ของ Windows ซึ่งเตรียมไว้ให้ผู้ใช้พร้อมใช้งาน ประกอบด้วย 3 files หลักดังนี้

- hid.dll ใช้สำหรับ connect อุปกรณ์ USB ที่กำลังแสดงตนเข้ามาใหม่
- setupapi.dll ใช้ในการหาลักษณะเฉพาะต่างๆ ของอุปกรณ์ เช่น Product Identification (PID),Vender Identification (VID)
- kernel32.dll ใช้ในการแลกเปลี่ยนข้อมูลกับอุปกรณ์



รูปที่ 1 การติดต่อระหว่าง Software กับ Hardware

เพื่อให้ผู้ใช้งานอุปกรณ์ เขียนโปรแกรมติดต่อกับอุปกรณ์ได้สะดวกยิ่งขึ้น Windows จึงเตรียม API Functions เพื่อดึงค่า Function ต่างๆใน DLL ทั้ง 3 files นี้มาใช้งาน เมื่อเรียกใช้ API Functions ในส่วนของ Software ของ Windows จะจัดการกับระบบเองโดยจะติดต่อกับ Kernel Mode จนถึง Hardware ซึ่งในที่นี้คือ Electronic Key แสดงในรูปที่ 1

ในการทำงานจริง API Functions ที่ถูกเรียกใช้จาก Windows ค่อนข้างจะเป็นวิธีเรียกใช้แบบทั่วไป แต่ในการส่งค่าไปให้ Device นั้น จำเป็นต้องทราบว่า Device ที่มีอยู่นั้น รับส่งข้อมูลในรูปแบบใด ซึ่งบริษัทผู้ผลิตอุปกรณ์จะให้ข้อมูลส่วนนี้ กับผู้ที่ต้องการนำเอาอุปกรณ์ไปพัฒนา จึงจะสามารถแลกเปลี่ยนข้อมูลให้กับ Device ได้

2.3 อุปกรณ์อินเตอร์เฟสพอร์ต USB

เพื่อให้ Host ติดต่อกับ Electronic Key โดยผ่านอุปกรณ์อินเตอร์เฟสตัวนี้ ซึ่งอุปกรณ์อินเตอร์เฟส USB มีคุณสมบัติดังนี้

- สนับสนุนการ Transfer mode ทั้ง Control และ Interrupt ซึ่ง Control mode ใช้สำหรับเป็นการติดต่อครั้งแรกที่มีการเสียบ Device เข้ากับ Host เพื่อให้ Host ทราบว่า Device ที่เสียบเข้ามาใน Host เป็น Device ที่อยู่ใน Class ใด ดังนั้นจึงจำเป็นจะต้องมี Control mode ในทุก ๆ USB Device สำหรับ Interrupt mode ใช้สำหรับในการวนอ่านข้อมูลใน Device
- สามารถอินเตอร์เฟสกับ Microcontroller ได้โดยผ่านทาง Parallel Port หรือแบบอื่นๆ
- ต้องการแรงดันขณะทำงานไม่เกิน 5 V และกินกระแสน้อยกว่า 500 mA ซึ่งเป็นข้อจำกัดของ USB Port
- มีขนาดเล็กเพื่อให้เหมาะกับงานประเภทนี้
- มี Endpoint ที่พร้อมใช้งานมากกว่า 2 Endpoint ซึ่ง Endpoint คือตำแหน่งสุดท้ายที่ Host ติดต่อมายัง Device ดังนั้นต้องใช้ 1 Endpoint ในการติดต่อแบบ Control mode เพื่ออ่านค่า Device และ 1 Endpoint สำหรับ Interrupt mode ในการรับส่งข้อมูล
- อุปกรณ์ที่มีลักษณะสอดคล้อง ตามที่กล่าวมาได้แก่ NetChip#NET2888 ,PIC#16C745 และ Philips#PDIUSBD11, D12 เป็นต้น

2.3.1 Microcontroller

ทำหน้าที่คำนวณตัวเลขที่เกิดจาก การเข้าและถอดรหัส พร้อมตรวจสอบความถูกต้องของ Electronic Key และเก็บข้อมูลสำหรับให้ Host วนมาอ่านเพื่อแสดงให้เห็นถึงผู้ใช้ต่อ Electronic Key เข้ากับระบบก่อนการใช้งานซึ่ง Microcontroller ที่ใช้ในการเก็บข้อมูลดังกล่าว จะต้องมีคุณสมบัติดังนี้

- มีขนาดของ Program Memory และ RAM ใหญ่
- ป้องกันการอ่านข้อมูลได้ ซึ่งเป็นผลให้ไม่สามารถทำ Electronic Key ซ้ำขึ้นมาใหม่ได้
- มีขนาดเล็ก ต้องการกระแสน้อยกว่า 500 mA
- อุปกรณ์ที่มีลักษณะสอดคล้อง ตามที่กล่าวมาได้แก่ PIC#16F87xA ,PIC#18F242

2.3.2 USB Microcontroller

เป็นอุปกรณ์ที่รวมเอาอุปกรณ์ที่กล่าวมาข้างต้น มารวมกัน ซึ่งลักษณะการทำงานก็จะคล้ายกัน เพียงแต่รวมเอาทั้งอุปกรณ์ USB Interface และ Microcontroller มารวมเข้าด้วยกันเพื่อประหยัดต้นทุนในการสร้าง ทั้งนี้อุปกรณ์ USB Microcontroller ที่ใช้ทำเป็น Electronic Key ต้องมีคุณสมบัติคล้ายคลึงกับที่กล่าวมาอุปกรณ์ USB ประเภทนี้ได้แก่ Cypress#CY7C637xx, ScanLogic#SL11R

2.4 การเข้ารหัสแบบ RSA [3]

อัลกอริทึม RSA ได้รับการพัฒนาขึ้นที่มหาวิทยาลัย MIT ในปี 1977 โดยศาสตราจารย์ 3 คน ซึ่งประกอบด้วย Ronald Rivest, Adi Shamir และ Leonard Adleman ชื่อของอัลกอริทึมได้รับการตั้งชื่อตามตัวอักษรตัวแรกของนามสกุล ของศาสตราจารย์ทั้งสามคน ซึ่งกำหนดให้ n = pq โดย p และ q เป็นจำนวนเฉพาะที่ไม่ซ้ำกันและเขียน Euler totient Function ได้เป็น

$$\phi(n) = (p-1)(q-1) \qquad (1)$$

เลือกตัวเลข $1<e<\phi(n)$ ได้เป็น

$$\gcd(e, \phi(n)) = 1 \qquad (2)$$

คำนวณค่า d จาก

$$d = e^{-1} \bmod \phi(n) \qquad (3)$$

โดยที่ e เป็น public exponent และ d เป็น private exponent โดยทั่วไปจะเลือกค่า e น้อยๆ เช่น $e = 2^{16}+1$ ส่วนค่าของ d ,p และ q จะต้องเป็นความลับเพื่อใช้ในการคำนวณการเข้ารหัสตามสมการ $C = M^e \pmod n$ เมื่อ M คือข้อมูลดิบซึ่งเป็นค่า $0 \le M < n$ ค่าของ C เป็นค่าที่เข้ารหัสแล้ว ซึ่งสามารถถอดรหัสได้จาก $M = C^d \pmod n$

ขั้นตอนการเข้าและถอดรหัส

1. สุ่มเลข จำนวนเฉพาะมา 2 ตัวในที่นี้กำหนดให้เป็น p,q
2. n = p*q
3. เลือกเลข จำนวนเฉพาะมา 1 ตัวคือ d ซึ่งต้องเป็นไปตามเงื่อนไขนี้ $1<d<\phi(n)$
4. หาค่าผกผันของ d คือ $ed \bmod \phi(n) = 1$

การเข้ารหัสสามารถทำได้โดย นำข้อมูลที่ต้องการเข้ารหัส เช่นต้องการเข้ารหัสคำว่า RENA โดยกำหนดให้ A=00, B=01, C=02,…,Z=25 จากนั้นทำตามขั้นตอนทั้ง 4 ข้างต้น คือ p = 53 และ q = 61 ได้ n = 3233 และ $\phi(n)$ = 3120 เลือก d = 791ได้ e = 71 นำค่าต่างๆ มาเข้ารหัสด้วยสมการ $C = M^e \bmod n$ ได้ RE=1704 และ NA=1300 จากนั้นแทนในสมการได้ $1704^{71} \bmod 3233 = 3106$ และค่า $1300^{71} \bmod 3233 = 0100$ นั้นหมายความว่าข้อมูลที่เข้ารหัสแล้วคือ 3106 0100 และในการถอดรหัสใช้สมการ $M = C^d \bmod n$ ได้ $3106^{791} \bmod 3233 = 1704$ ซึ่งเป็นข้อมูลที่ถูกต้อง

ในการนำไปใช้กับ Electronic Key ไม่สามารถนำวิธีการนี้ไปใช้ได้โดยตรง เนื่องจากจำนวนของตัวเลขที่คำนวณมีค่ามากเช่น $21^{31} = 9.74536560714604461109210780048 87e+40$ สังเกตว่าตัวเลขมากเกินกว่าที่ Microcontroller จะคำนวณได้ ดังนั้นจึงดำเนินการศึกษาสมการรูปแบบ Modular Exponentiation

2.5 การลดทอนสมการรูปแบบ RSA [4]

การคำนวณสมการการเข้ารหัสแบบ RSA โดยทั่วไปจะถูกเรียกว่า Modular Exponentiation ซึ่งจะประกอบด้วย exponentiation คือ M^e และส่วนที่เป็นการ Modular คือ C := (M^e) mod n ผลการคำนวณในแต่ละขั้นตอน เพื่อหาผลลัพธ์จำเป็นต้องลดลง ในการคำนวณวิธีการต่างๆ ซึ่งจะไม่ยกกำลังสมการโดยตรงด้วยวิธีของ Montgomery reduction algorithm ถูกค้นคิดพัฒนาเพื่อให้เหมาะกับ Signal processors หรือ Microprocessors ซึ่งจะมีความสามารถในการประมวลผลทางคณิตศาสตร์ modulo กำลังของ 2 ได้รวดเร็ว ดังนั้นการประมวลผลของ Montgomery reduction algorithm จะปราศจากการนำค่ามา modulo n โดยตรง แต่จะทำการแปลง n ให้อยู่ในรูปของ 2 ยกกำลัง n นั้นคือ r กำหนดให้ $2^{k-1} \leq n < 2^k$ เมื่อ r เป็น 2^k โดยที่ r และ n จะต้องเป็น relative prime เช่น gcd(2^k,n)=1 แนวคิดของ Montgomery reduction algorithm จะอาศัยพื้นฐานดังนี้

ให้จำนวนเต็มบวก a < n จะได้

$$\overline{a} = a \cdot r \bmod n \qquad (4)$$

Montgomery product ต้องการค่า $\overline{a}$ และ $\overline{b}$

$$\overline{R} = \overline{a} \cdot \overline{b} \cdot r^{-1} \bmod n \qquad (5)$$

ค่า r^{-1} เป็นค่า inverse ของ r modulo n เช่น

$$r^{-1} \cdot r = 1 \bmod n \qquad (6)$$

ผลของจำนวนจริงเขียนได้ดังนี้

$$R = a \cdot b \bmod n \qquad (7)$$

เมื่อ $\overline{R} = \overline{a} \cdot \overline{b} \cdot r^{-1} \bmod n$ จะได้

$$= a \cdot b \cdot r \cdot r^{-1} \bmod n$$

$$= a \cdot b \cdot r \bmod n \qquad (8)$$

Montgomery reduction algorithm ต้องการค่า n' ซึ่งเป็นจำนวนจริงจากสมการ

$$r \cdot r^{-1} - n \cdot n' = 1 \qquad (9)$$

จำนวนจริง r^{-1} และ n' ทั้งสองค่านี้สามารถหาได้จาก extended Euclidean algorithm ซึ่ง Montgomery product algorithm จะคำนวณหาค่า u = $\overline{a} \cdot \overline{b} \cdot r^{-1} \bmod n$ โดยใช้ function MonPro($\overline{a}, \overline{b}$) ดังนี้

Function MonPro($\overline{a}, \overline{b}$)

Step 1. t: = $\overline{a}, \overline{b}$

Step 2. m: = t * n' mod r

Step 3. u: = (t + m*n)/r

Step 4. if u $\geq$ n then return u-n else return u

ลักษณะที่สำคัญที่สุดของ Montgomery product algorithm คือการประมวลผลโดย modulo r และ division r ซึ่งจุดเด่นทั้งสองการประการนี้ สามารถทำงานได้รวดเร็วเนื่องจาก r คือ 2^k

นอกจากต้องใช้ function Montgomery product แล้วยังจำเป็นต้องใช้ function modular exponentiation ซึ่งเป็น function หลักในการหาเข้ารหัส โดย function นี้จะใช้หลักการพื้นฐานของ binary method [5]

Function ModExp(M,e,n) {n is odd}

Step 1. Compute n' using Euclid's algorithm

Step 2. $\overline{M}$ = M*r mod n

Step 3. $\overline{C}$ = r mod n

Step 4. for i = k-1 down to o do

Step 4a. $\overline{C}$ = MonPro($\overline{C}$, $\overline{C}$)

Step 4b. if e_i=1 then $\overline{C}$ = MonPro($\overline{M}$, $\overline{C}$)

Step 5. C = MonPro($\overline{C}$,1)

Step 6. return C

จาก function ModExp(M,e,n) จะเริ่มโดยการหาค่า $\overline{M}$ โดยการใช้ตัวหาร ซึ่งมีลักษณะคล้ายๆ กับใน Step 2 และ 3 ของ function MonPro($\overline{a}, \overline{b}$) และไม่จำเป็นต้อง Preprocessing ก่อนการคำนวณจริงเนื่องจากมันถูกคำนวณในลูปของ Montgomery product ซึ่งก็คือ modulo 2^k และ division 2^k

จากหลักการที่กล่าวมาข้างต้นสามารถเขียนขั้นตอนการหาค่า C: = M^e mod n ได้ดังนี้

1. r = 2^k โดยหา k ได้จาก $2^{k-1} \leq n < 2^k$

   เมื่อ k คือจำนวนบิตของ n

   n คือ p x q

2. นำค่า r และ n หา gcd(r,n) = 1 โดยมี Algorithm ในการหาดังนี้

GCD Algorithm

INPUT: two positive integers xand ywith x $\geq$ y

OUTPUT:gcd (x,y)

1.g=1

2. While both x and y are even do the follow in g : x=x/2 , y=y/2 , g=2g

3. While x $\neq$ 0 do the following:

   3.1. While x is even do : x=x/2

   3.2. While y is even do : y =y/2

   3.3. T= | x-y| /2

   3.4. If x $\geq$ y then x=t; otherwise, y=t

4.Return (g*y)

3. ใช้ Extended Euclid algorithm หาค่า r^{-1} และ n' จากสมการ $r*r^{-1} - n*n' = 1$

Binary extended gcd algorithm

1. g = 1
2. While x and y are both even, do the following : x=x/2 , y=y/2 , g=2g
3. u=x , v=y , A=1 , B=0 , C=0 , D=1
4. While u is even do the following:
   4.1 u = u / 2
   4.2. if A=B=0 (mod 2) then A=A/2,B=B/2 ; otherwise, A=(A+y) / 2 , B=(B-x)/2
5. While v is even do the following
   5.1 v = v/2
   5.2. if C=D=0 (mod 2) then C=C/2 , D=D/2 ; otherwise, C=(C+y)/2,D=(D-x)/2
6. If u > v then u = u – v , A= A-C ,B=B-D; otherwise, v = v–u , C = C- A ,D=D-B
7. If u = 0, then a = C, b=D and return (a,b,g*v); otherwise, go to step 4

4. ใช้ function MonPro($\overline{a},\overline{b}$) และ function ModExp(M,e,n) เพื่อหาค่า C จากสมการ $C = M^e \bmod n$ โดยที่ e ต้องแปลงเป็นฐานสองซึ่งใช้วิธีการของ Binary method

ตัวอย่างในการคำนวณหาค่า $C = 7^{10} \bmod 13$ โดยใช้ Montgomery reduction algorithm

<u>วิธีทำ</u> M = 7, e= 10, n=13

ได้ $r = 2^k = 16$ ใช้ Euclid's algorithm หาค่า r^{-1} และ n' ได้ 16*9 – 13*11 = 1 นั้นคือ $r^{-1} = 9$ และ n' = 11

หา $\overline{M}$ จาก M = 7

$\overline{M}$ =M*r mod n = 7*16 mod 13 = 8

หา $\overline{C}$ จาก

$\overline{C}$ = r mod n = 16 mod 13 = 3

ดังนั้นได้ $\overline{C}$ = 3 และ $\overline{M}$ = 8

ตารางที่ 1 ตัวอย่างการคำนวณ Function MonPro

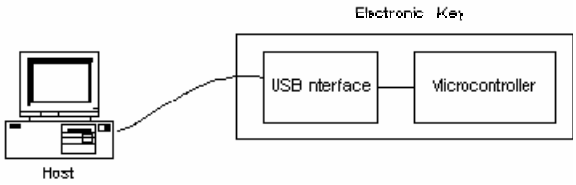
e_i	Step 5	Step 6
1	MonPro(3,3) =3	MonPro(8,3)=8
0	MonPro(8,8)=4	
1	MonPro(4,4)=1	MonPro(8,1)=7
0	MonPro(7,7)=12	

Step 7: C= MonPro(12,1) = 4

จากตัวอย่างข้างต้น หากคำนวณค่าโดยไม่ใช้ทฤษฎีของ Montgomery reduction algorithm จะได้ 7^{10} = 282 475 249 จะเห็นได้ว่าตัวเลขจะมากขึ้นเป็นทวีคูณซึ่งเป็นผลมาจากการยกกำลังตัวเลขและหากเลขยกกำลังมีค่ามาก ผลลัพธ์ของการยกกำลังจะมากทวีคูณตามไปด้วย ซึ่งเมื่อใช้ Montgomery reduction Algorithm พบว่าค่าที่มากที่สุดจะเกิดขึ้นใน Step 2 และ Step 3 ของ Function Monpro = 99 ซึ่งจะเห็นได้ว่าตัวเลขที่เกิดจากการคำนวณแตกต่างกันหลายเท่า เมื่อใช้ Montgomery reduction Algorithm เข้าช่วยตัวเลขที่ยกกำลังมากๆ จะทำให้การวนทำซ้ำของ function Monpro มีรอบทำงานหลายรอบ (เท่ากับค่าของ e) แต่ไม่มีผลให้ตัวเลขที่เกิดจากการทำซ้ำมากขึ้น ซึ่งมีความเป็นไปได้ที่การเข้ารหัสแบบ RSA สามารถทำได้บน Microcontroller เนื่องจาก Microcontroller ไม่สามารถคำนวณตัวเลขขนาดใหญ่ได้

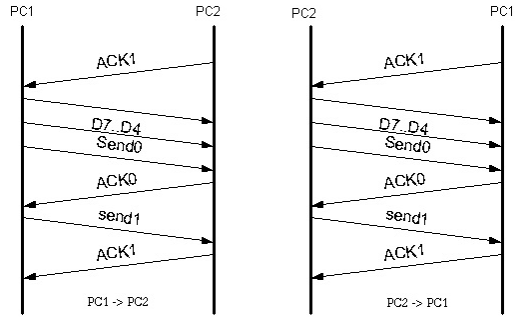
3. การออกแบบกุญแจอิเล็กทรอนิกส์

โครงสร้างของกุญแจอิเล็กทรอนิกส์ประกอบด้วย 2 ส่วนคือ USB Interface และ Microcontroller แสดงในรูปที่ 2 โดยการติดต่อสื่อสารกันผ่านทาง Parallel Port ซึ่งการสื่อสารจะเกิดขึ้นได้จะต่อเมื่อมีข้อตกลงในการสื่อสารหรือเรียกว่า Protocol โดยในการรับส่งข้อมูล ฝ่ายรับกับฝ่ายส่งต้องใช้ Protocol เดียวกัน ดังนั้น



รูปที่ 2 โครงสร้างกุญแจอิเล็กทรอนิกส์

ก่อนที่จะมีการส่งข้อมูลระหว่าง PC ต้องกำหนดให้ PC1 รอสัญญาณ ACK1 จาก PC2 ก่อน นั้นหมายความว่าหาก PC2 พร้อมรับข้อมูล PC2 จะส่ง ACK1 มาให้ PC1 เมื่อ PC1 ได้รับ ACK1 กลับมา PC2 จะส่ง DATA ที่มีขนาด 4 บิตคือ D7..D4 ขณะนี้ PC2 ยังไม่รับข้อมูล โดยจะรับข้อมูลเมื่อได้รับ Send0 จาก PC1 แล้วเท่านั้น เมื่อ PC2 รับข้อมูลเรียบร้อยแล้วจะส่ง ACK0 ออกไปที่ PC1 เพื่อแจ้งให้ PC1 ทราบว่าขณะนี้ได้รับข้อมูลเรียบร้อยแล้วและ PC1 จะส่ง send1 เพื่อแจ้งให้ PC2 ทราบว่าขณะนี้ PC1 ได้รับทราบแล้วว่า PC2 ได้รับข้อมูลแล้ว ดังนั้น PC2 จะ ACK1 กลับไปที่ PC1 ว่าพร้อมที่จะรับข้อมูลในชุดต่อไปแล้ว ซึ่งข้อมูลในชุดต่อไปก็จะเป็นไปตามทำนองเดียวกันนี้ ดังแสดงในรูปที่ 3



รูปที่ 3 Protocol ในการสื่อสาร

4. การประยุกต์ใช้งาน

เมื่อดำเนินการออกแบบให้ Electronic Key ทำงานได้ตามที่ต้องการแล้ว สิ่งที่ต้องพิจารณาอีกประการหนึ่งคือความสะดวกในการนำไปประยุกต์ใช้งาน เมื่อนำไปใช้งานจริง Programmer จะต้องนำ Source Code บางส่วนที่เขียนขึ้นด้วย Visual Basic ของงานวิจัยนี้ไปรวมกับ Source Code ของโปรแกรมที่ต้องการจำกัดสิทธิ์การใช้งาน ดังนั้นเพื่อความสะดวกในการใช้งานและรักษาความปลอดภัยของ Source Code ในระบบ จึงต้องแยก Source Code ออกเป็น 2 ส่วนคือ ส่วนที่เป็น DLL (Dynamic Link Library) ซึ่งเป็นส่วนที่เข้าและถอดรหัส รวมถึงการตรวจสอบ ID ของ Electronic Key ทำให้ Source Code มีความปลอดภัยจากผู้นำไปประยุกต์ ส่วนที่ 2 คือ Source Code ที่ใช้ในการติดต่อระหว่าง Host กับ Electronic Key จะอยู่ภายนอก DLL เมื่อ Programmer นำไปใช้งานสามารถเขียนโปรแกรมเพื่อตรวจสอบ Electronic Key ได้จากส่วนนี้

5. สรุป

หลังจากที่ผู้วิจัยได้ทำการออกแบบและสร้างกุญแจอิเล็กทรอนิกส์แล้วสามารถสรุปผลการวิจัยได้ดังนี้

- กรณีผู้บุกรุกกระทำการดักจับข้อมูลขณะที่ Host กำลังสื่อสารกับ Electronic Key ผู้บุกรุกจะไม่สามารถตีความข้อมูลที่ดักจับได้ เนื่องจากข้อมูลถูกเข้ารหัสแบบ RSA ทำให้ข้อมูลเปลี่ยนแปลงทุกครั้งที่มีการสื่อสาร
- กรณีผู้บุกรุกต้องการทำซ้ำ (Copy) Electronic Key จะไม่สามารถทำได้เนื่องจากในขณะสร้าง Electronic Key ผู้วิจัยได้ทำการป้องการการอ่านข้อมูลภายใน Electronic Key แล้ว
- กรณีผู้บุกรุกสร้าง Electronic Key ขึ้นมาเองโดยเลียนแบบ Electronic Key ของระบบเดิมจะไม่สามารถทำได้เนื่องจากใน Electronic Key ของระบบมีการฝังค่า ID ไว้ภายในซึ่งค่านี้จะเก็บไว้เป็นความลับผู้บุกรุกไม่สามารถรู้ได้

กิตติกรรมประกาศ

งานวิจัยนี้ได้รับการสนับสนุนจาก บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่

เอกสารอ้างอิง

[1] V.Gadre.Dhananjay, 2000. Programming and customizing the AVR Microcontroller, New York USA.

[2] Axelson Jan, 2001.USB Complete Everything You Need to Develop Custom USB Peripherals, WA USA.

[3] Schneier Bruce, 1996 .Applied Cryptography Second Edition: protocols, algorithms, MN USA.

[4] RSA Laboratories. 1994, High-Speed RSA Implementation. RSA Data Security, Inc. 100 Marine Parkway, Suite 500 Redwood City, CA

[5] D.E Knuth, 1981.The Art of Computer Programming: Seminumerical Algorithms, Volume 2. Reading, MA: Addison-Wesley,Second edition.