

เอกสารประกอบชิ้นงาน

รายวิชา Digital System Fundamentals รหัสวิชา 1076006

Smart lock box

สมาชิก

1.นายธัญธร พรสวัสดิ์ชัย รหัสนักศึกษา 61010497 Sec เช้า

2.นายนพณัฐ พันลุตัน รหัสนักศึกษา 61010541 Sec เช้า

3.นายพิพิธพงศ์ จิตภักดิ์ไทย รหัสนักศึกษา 61010750 Sec เช้า

4.นายธนธัช อัศวรัตนากร รหัสนักศึกษา 62015048 Sec บ่าย

5.นายพีรวัส อ่ำโต รหัสนักศึกษา 62015087 Sec บ่าย

นำเสนอ

รศ. ดร. เจริญ วงษ์ชุ่มเย็น

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Idea/Useful | Techniques | Completeness | Report/Clip | Present |
|  |  |  |  |  |

**คำนำ**

ชิ้นงานนี้เป็นส่วนหนึ่งของรายวิชา Digital System Fundamentals รหัสวิชา 1076006 โดยมีจุดประสงค์เพื่อนำเอาความรู้ที่ได้จากรายวิชา Digital System Fundamentals มาประยุกต์ใช้ในชิ้นงานนี้ โดยมีความสำคัญในการนำความรู้ที่ได้รับมาใช้ควบคู่กับความคิดสร้างสรรค์ของผู้จัดทำเพื่อนำเสนอชิ้นงานซึ่งมีความเหมาะสมแก่ความรู้ซึ่งได้รับจากรายวิชานี้

ซึ่งในเอกสารประกอบชิ้นงานนี้จะมีเนื้อหาเกี่ยวกับรายละเอียดในการออกแบบการทำงานของระบบดิจิตอลในการทำ Smart lock box โดยใช้ FPGA รวมถึง TTL ที่ใช้ในการจัดทำชิ้นงานนี้

ทั้งนี้ทางคณะผู้จัดทำหวังเป็นอย่างยิ่งว่าเอกสารนี้จะเป็นประโยชน์แก่ผู้ที่สนใจที่จะศึกษาเกี่ยวกับชิ้นงานที่คณะผู้จัดทำได้จัดทำขึ้นไม่มากก็น้อย หรือใช้เป็นแรงบัลดาลใจไปต่อยอดในชิ้นงานอื่นๆได้

คณะผู้จัดทำ

**สารบัญ**

[**บทที่ 1 ที่มา หลักการ และเหตุผล** 4](#_Toc27319694)

[**ที่มาและความสำคัญ** 4](#_Toc27319695)

[**วัตถุประสงค์ของชิ้นงาน** 4](#_Toc27319696)

[**บทที่ 2 ทฤษฎีที่เกี่ยวข้อง** 5](#_Toc27319697)

[**บทที่ 3 การดำเนินการชิ้นงาน** 11](#_Toc27319698)

[รายละเอียดของชิ้นงาน 11](#_Toc27319699)

[กระบวนและเทคนิคการออกแบบ 11](#_Toc27319700)

[วัสดุอุปกรณ์ที่ใช้ 21](#_Toc27319701)

[ขั้นตอนการดำเนินงาน 21](#_Toc27319702)

[**บทที่ 4 ผลการดำเนินงาน** 22](#_Toc27319703)

[วิธีการใช้ชิ้นงาน 22](#_Toc27319704)

[Source code : 22](#_Toc27319705)

[**บทที่ 5 สรุปผลการดำเนินงาน** 32](#_Toc27319706)

[สรุปผลการดำเนินงาน 32](#_Toc27319707)

[ปัญหา อุปสรรค และการแก้ไข 32](#_Toc27319708)

[**แหล่งอ้างอิง** 33](#_Toc27319709)

# **บทที่ 1 ที่มา หลักการ และเหตุผล**

## **ที่มาและความสำคัญ**

ปัจจุบันนี้ความเป็นส่วนตัวและความปลอกภัยเป็นเรื่องที่ใครๆก็ต้องการ หากเรานึกถึงที่เก็บของที่มีความเป็นส่วนตัวและปลอดภัย เราก็จะนึกถึงตู้เซฟกัน แต่ทว่าตู้เซฟปกตินั้นยากต่อการพกพาไปไหนมาไหน และลำบากเวลาที่เราจะเปิดตู้ด้วย ดังนั้น Smart Lock จึงเข้ามามีความสำคัญ ด้วยขนาดที่สามารถพกพาไปที่อื่นๆได้ รวมกับระบบป้องกันความปลอดภัย 2 ชั้น RFID และ Password

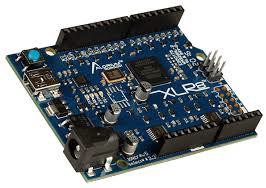
## **วัตถุประสงค์ของชิ้นงาน**

1.เพื่อให้ผู้ใช้ทำกิจกรรมระหว่างวันได้โดยไม่ต้องกังวลเรื่องสัมภาระ

2.เพื่อเพิ่มความสบายใจให้แก่ผู้ใช้ด้วยระบบล็อค 2 ชั้น ด้วยรหัสผ่านและบัตร RFID

# **บทที่ 2 ทฤษฎีที่เกี่ยวข้อง**

**FPGA**

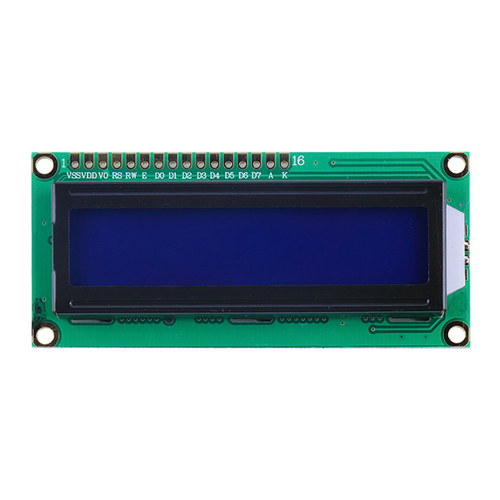
เอฟพีจีเอ หรือ อุปกรณ์ลอจิกแบบโปรแกรมได้ (อังกฤษ: field programmable gate array: FPGA) บางครั้งอาจจะมีคนสับสนกับคำว่า flip-chip pin grid array ซึ่งมีโครงสร้างตัวถังคล้ายๆ กัน

FPGA จัดเป็น อุปกรณ์สารกึ่งตัวนำชนิดโปรแกรมได้ที่มีโครงข่ายการเชื่อมต่อภายในแบบแมตริกซ์ โครงสร้างภายในของ FPGA นั้นสามารถโปรแกรมให้มีหน้าที่การทำงานเหมือนลอจิกเกตพื้นฐาน เช่น AND, OR, XOR, NOT หรือรวมกันหลายๆ ชนิด (combinational logic) เพื่อให้ทำหน้าที่ที่มีความซับซ้อนเพิ่มขึ้น เช่น decoders หรือฟังก์ชันทางตณิตศาสตร์ ใน FPGAs ทั่วไป นอกจากจะประกอบด้วยส่วนของวงจรลอจิกแบบโปรแกรมได้แล้ว จะยังมีบล็อกของหน่วยความจำ ซึ่งอาจจะสร้างด้วยฟลิบฟลอปอย่างง่าย หรือใช้พื้นที่ของสารกึ่งตัวนำสร้างเป็นหน่วยความจำจริงๆ อยู่ภายในก็ได้

ในการออกแบบวงจรดิจิตอลอิเล็กทรอนิกส์ ที่มี FPGA อยู่บนแผงวงจรด้วยนั้น จะช่วยให้ผู้ออกแบบสามารถลดขนาดของแผงวงจร รวมทั้งสามารถออกแบบได้รวดเร็ว ไม่ต้องทดสอบรายละเอียดภายในให้เสร็จสมบูรณ์ 100 % ก็สามารถออกแบบแผงวงจรได้ เมื่อได้รับแผงวงจรและประกอบอุปกรณ์ต่างๆ เสร็จแล้ว จึงค่อยกำหนดหน้าที่การทำงานของ FPGA ได้ในภายหลัง ต่างจากการออกแบบด้วยลอจิกเกตขนาดเล็ก ที่ต้องออกแบบทางเดินของลายทองแดงให้เสร็จสมบูรณ์ก่อน และไม่สามารถแก้ไขได้ในภายหลัง นอกจากนี้ การใช้งาน FPGA สามารถโปรแกรมการทำงานได้ในทุกขณะแม้แต่ขณะที่ส่งมอบงานแล้ว ก็ยังสามารถเข้าไปแก้ไขวงจรได้โดยง่ายดาย จึงเป็นที่มาของคำว่า "field programmable" ซึ่งก็หมายถึงโปรแกรมได้ในภาคสนามหรือที่หน้างานนั่นเอง อย่างไรก็ตามข้อกำหนด (Configuration) ของ FPGA จะหายไปหลังจากปิดไฟเลี้ยง ดังนั้น จะต้องมีหน่วยความจำภายนอก (Flash) มาคอยรักษาข้อกำหนดของ FPGA ไว้ ซึ่ง FPGA จะมีกระบวนการอ่านข้อกำหนดนั้นโดยอัตโนมัติหลังจากได้รับไฟเลี้ยง

การทำงานของ FPGAs จะยังมีความเร็วที่ด้อยกว่าapplication-specific integrated circuit (ASIC) , และเมื่อเปรียบเทียบขนาดทางกายภาพ พบว่าจะมีความหนาแน่นของวงจรที่น้อยกว่า รวมทั้งใช้กำลังงานมากกว่า ASIC อย่างไรก็ตาม FPGA มีข้อได้เปรียบตรงที่ใช้เวลาในการพัฒนาผลิตภัณฑ์ (time to market) ที่น้อยกว่า สามารถแก้ไขวงจรได้หลังจากที่ใช้งานจริงในภาคสนาม ,และมีค่าแรงในการดำเนินการที่ต่ำกว่า (non-recurring engineering) . นอกจากนี้ ยังมี FPGA ชนิดที่โปรแกรมได้ครั้งเดียว (OTP) ซึ่งมีราคาที่ต่ำกว่าโดย FPGA ชนิดนี้เมื่อโปรแกรมแล้วจะคล้ายกับ ASIC นอกจากนี้ยังมีการรวมหน่วยความจำ config เข้าไว้ในอุปกรณ์ FPGA ซึ่งจะยังคงอยู่แม้ปิดไฟเลี้ยง เรียกว่า Complex programmable logic devices

**LCD 16x2**



LCD หรือ Liquid Crystal Display คือหน้าจอแสดงผลตัวอักษร ตัวเลขหรืออักขระต่างๆ (Character LCD) รวมถึงบางรุ่นที่สามารถแสดงภาพกราฟฟิคได้ด้วย (Graphic LCD)

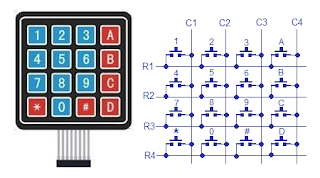
จอ LCD นี้จำเป็นมาก สำหรับงานที่ต้องการแสดงผลการทำงานต่างๆ เพื่อให้ผู้ใช้งานสามารถเชื่อมเป็นส่วนหนึ่งกับวงจรอิเล็กทรอนิกส์นั้นๆได้

**RFID**



RFID ย่อมาจาก Radio Frequency Identification เป็นระบบระบุลักษณะของวัตถุด้วยคลื่นความถี่วิทยุที่ได้ถูกพัฒนามาตั้งแต่ปี ค.ศ. 1980 มีวัตถุประสงค์หลักเพื่อนำไปใช้งานแทนระบบบาร์โค้ด (Barcode) โดยจุดเด่นของ RFID อยู่ที่การอ่านข้อมูลจากแท็ก (Tag) ได้หลายๆ แท็กแบบไร้สัมผัสและสามารถอ่านค่าได้แม้ในสภาพที่ทัศนวิสัยไม่ดี ทนต่อความเปียกชื้นแรงสั่นสะเทือน การกระทบกระแทก สามารถอ่านข้อมูลได้ด้วยความเร็วสูง โดยข้อมูลจะถูกเก็บไว้ในไมโครชิปที่อยู่ในแท็ก

**Keypad**



Keypad 4x4เป็นการจัดสวิทช์กดติดปล่อยดับจำนวน 16 ปุ่มในรูปแบบเมตริกซ์ 4 X 4 โดยปกติจะเป็นปุ่มสำหรับหมายเลข 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 และตัวอักษร A, B, C, D, \*, # Keypad 4x4จะมี 8 การเชื่อมต่อสายสัญญาณ คือ R1, R2, R3, R4 และ C1, C2, C3, C4 เป็นตัวแทนของแถวและคอลัมน์ตามลำดับ

**Servo**



เซอร์โวมอเตอร์ (Servo Motor) เป็นมอเตอร์ที่มีการควบคุมการเคลื่อนที่ของมัน (State) ไม่ว่าจะเป็นระยะ ความเร็ว มุมการหมุน โดยใช้การควบคุมแบบป้อนกลับ (Feedback control) เป็นอุปกรณ์ที่สามารถควบคุมเครื่องจักรกล หรือระบบการทํางานนั้นๆ ให้เป็นไปตามความต้องการ เช่น ควบคุมความเร็ว (Speed), ควบคุมแรงบิด (Torque), ควบคุมแรงตําแหน่ง (Position), ระยะทางในการเคลื่อนที่(หมุน) (Position Control) ของตัวมอเตอร์ได้ ซึ่งมอเตอร์ทั่วไปไม่สามารถควบคุมในลักษณะงานเบื้องต้นได้ โดยให้ผลลัพธ์ตามความต้องการที่มีความแม่นยําสูง

**Arduino**

Arduino เป็นชื่อเรียกของ platform micro controller ชนิดหนึ่งซึ่งก็ตามความหมายครับ micro=เล็ก, controller=ชุดควบคุม ดังนั้นหน้าที่ของมันก็คือชุดควบคุมขนาดเล็กที่สามารถนำไปเชื่อมต่อเพื่อสั่งการเครื่องมืออิเล็กทรอนิกส์ต่างๆได้นั่นเอง

สาเหตุหลักๆที่ทำให้ Arduino มีชื่อเสียงโด่งดัง อีกทั้งยังนับได้ว่าราคาย่อมเยามากกว่าบรรดา micro controller อื่นๆในตลาดก็คือการที่มันเป็น Open Source ทั้งในเชิง software และ hardware นั่นหมายความว่าใครๆก็สามารถนำระบบของ Arduino ไปดัดแปลงแก้ไขแล้วนำออกมาขายในตลาดได้โดยไม่ผิดกฎหมายครับ

Arduino ในส่วนของ Software



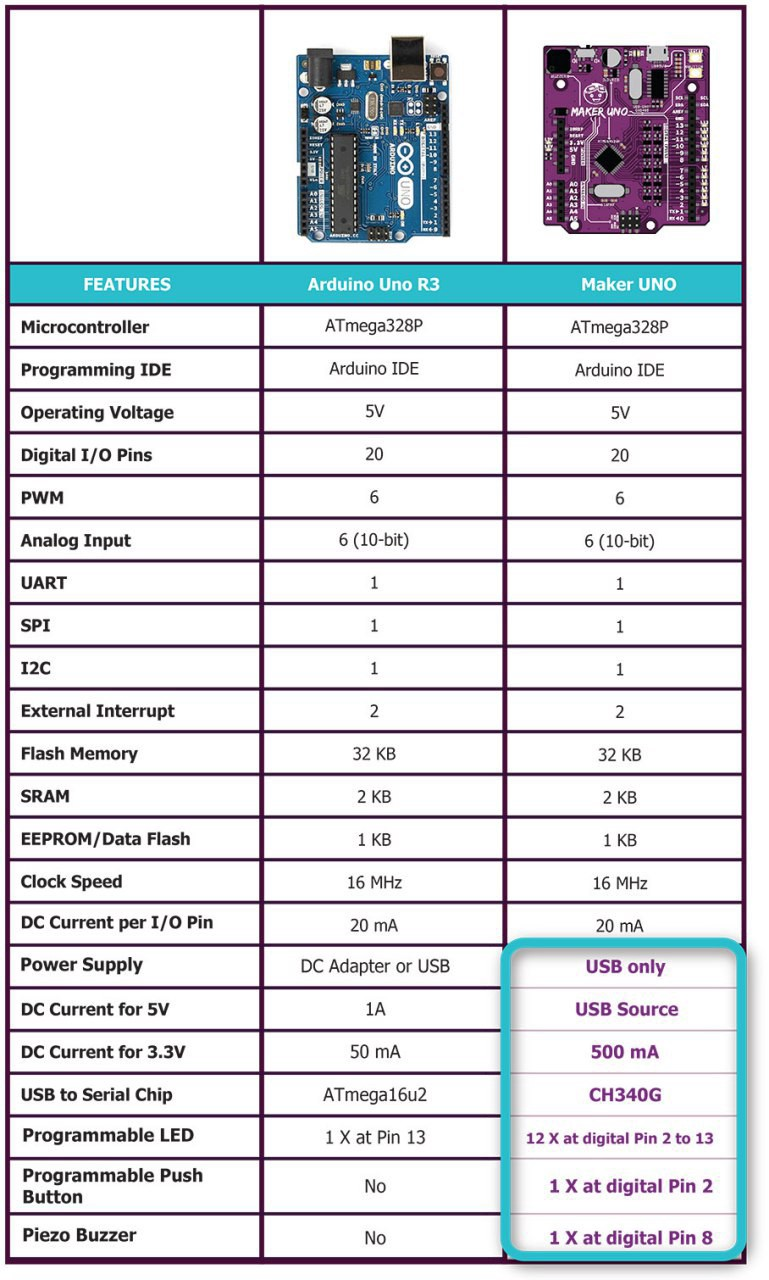
Arduino ได้พัฒนาส่วนของระบบการเขียนโปรแกรม หรือที่เรียกว่า IDE (Integrated Development Environment) ซึ่งจะช่วยให้เราฝังคำสั่งลงในบอร์ด Arduino ชนิดต่างๆได้ ซึ่งในส่วนของ software นี้ เรายังสามารถดาวโหลด library เพิ่มได้จาก internet ซึ่งทำให้เราเขียนโปรแกรมกับ controller ชนิดอื่นๆ ไม่จำกัดอยู่แค่ Arduino อีกด้วย

Arduino ในส่วนของ hardware

Arduino board แบ่งออกเป็นหลายชนิดด้วยกัน ซึ่งจะมีคุณลักษณะเหมาะสมกับการใช้งานไปคนละแบบ



**Maker Uno**



บอร์ด Maker UNO จาก Cytron เป็นบอร์ดไมโครคอนโทรเลอร์ ATmega328 ที่ถูกออกแบบมาให้ใช้งานง่าย ใช้ชิพ USB to serial CH340G พร้อมให้ on board LED เพิ่มมาถึง 12 ดวง และยังมี Push Button กับ Piezo Buzzer แบบโปรแกรมคุมคุมได้ เสริมมาให้ เพื่อให้ใช้งานง่ายยิ่งขึ้น

Technical Specifications

• SMD ATmega328P microcontroller

• USB Programming facilitated by the CH340.

• Input voltage: USB 5V, from computer, power bank or standard USB adapter.

• 500mA (maximum) 3.3V voltage regulator.

• 0-5V outputs with 3.3V compatible inputs.

• 14 Digital I/O Pins (6 PWM outputs).

• 6 Analog Inputs.

• ISP 6-pin Header.

• 32k Flash Memory.

• 16MHz Clock Speed.

• R3 Shield Compatible.

• LED array for 5V, 3.3V, TX, RX and all digital pins.

• On board programmable push button (pin 2, need to configure as INPUT\_PULLUP).

• On board piezo buzzer (pin 8).

• Utilize USB Micro-B socket.

• PURPLE PCB!

Dimension: 69.09mm(L) x 60.96(W)

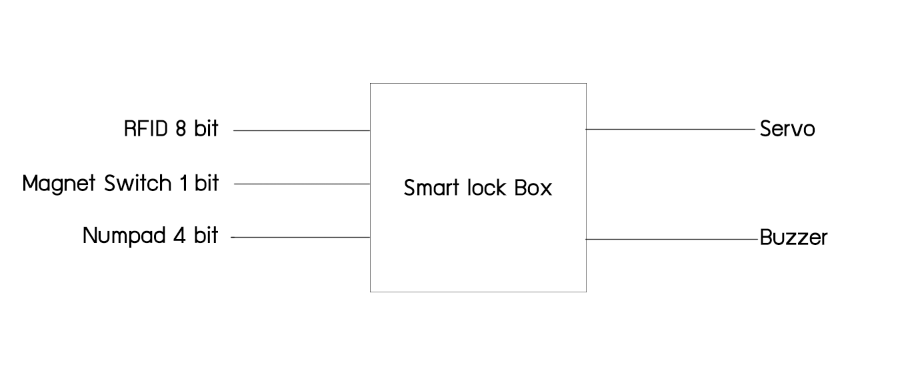
# **บทที่ 3 การดำเนินการชิ้นงาน**

## รายละเอียดของชิ้นงาน

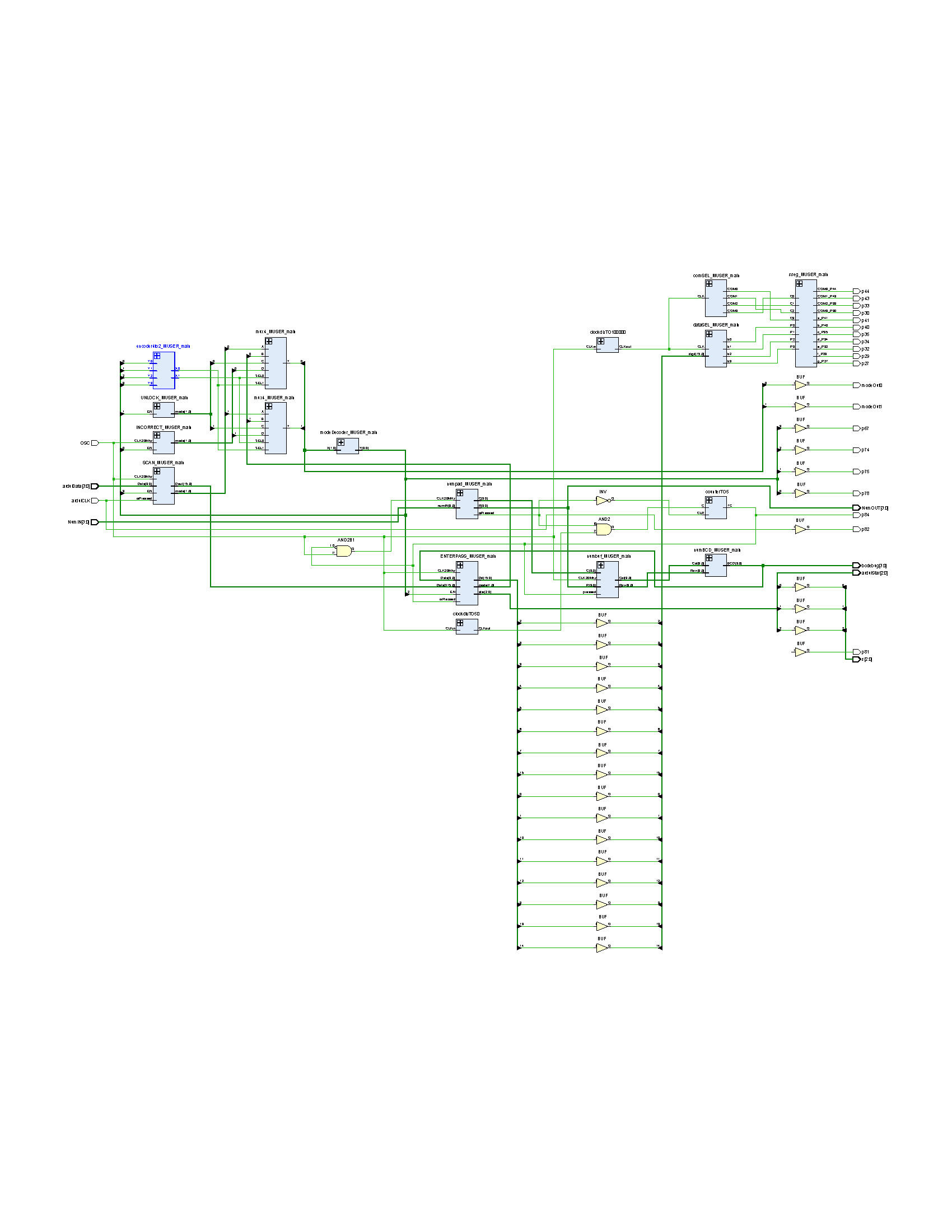
1. LCD แสดงสถานะของการทำงานตามการใช้งาน คือ Lock และ Unlock
2. สถานะการทำงานเปลี่ยน เมื่อมีการกระทำกับ Numpad หรือ แตะการ์ด RFID
3. มีเสียงเตือนดัง เมื่อมีการเปิดกล่อง Smart lock box อย่างผิดปรกติ
4. หากแตะการ์ด RFID และป้อนรหัสผ่านถูกต้อง Smart lock box จะเปลี่ยนสถานะ และกล่องถูกเปิดออก

## กระบวนและเทคนิคการออกแบบ

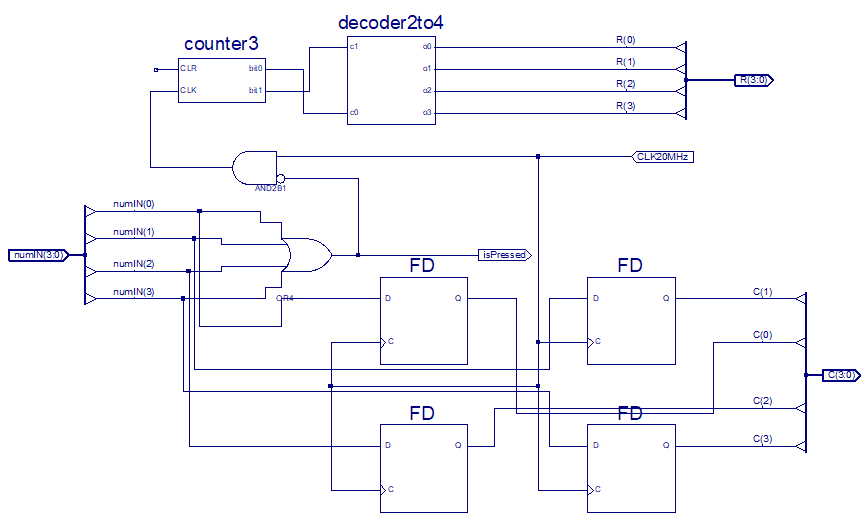
1. ใช้อุปกรณ์อ่าน RFID มารับค่าบัตร RFID เพื่อแสดงผลไปที่ LCD
2. ใช้ Magnet switch ในการรับค่าการ เปิด-ปิด ของกล่อง Smart lock box
3. ใช้ LCD ในกานแสดงสถานะของกล่อง Smart lock box
4. สร้างกล่อง Smart lock box จากกล่องไม้ โดยมีการเจาะเพื่อติดตั้งแผงป้อนข้อมูล
5. มีเสียงจาก Buzzer หากสถานะ Lock แต่มีการเปิดกล่องออก



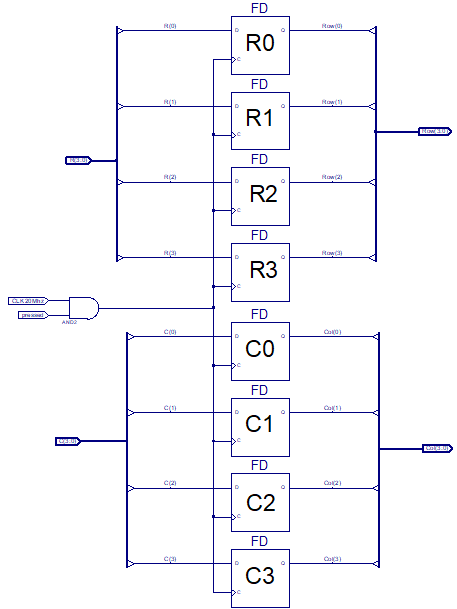
รูปที่ 1 Top-down ของวงจร



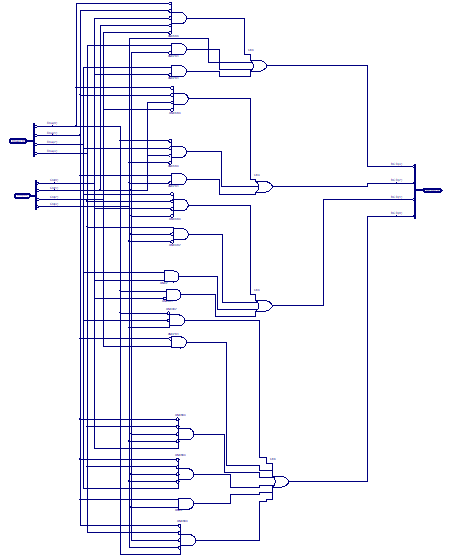
รูปที่ 2 แสดงแบบจำลองของวงจรทั้งหมด



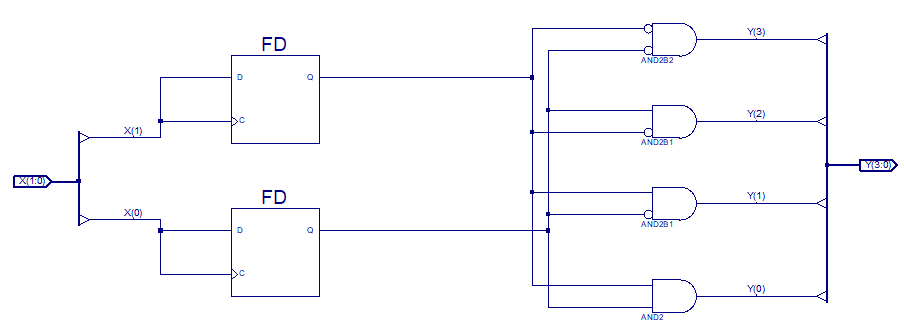
รูปที่ 3 แสดงวงจร Numpad



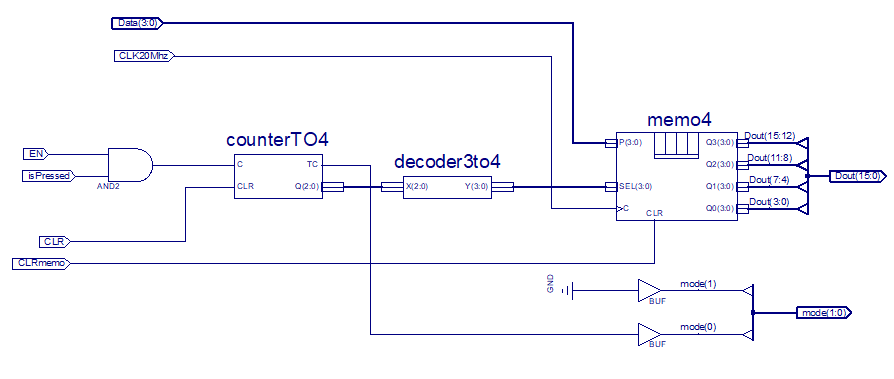
รูปที่ 4 แสดงวงจร Numbuf



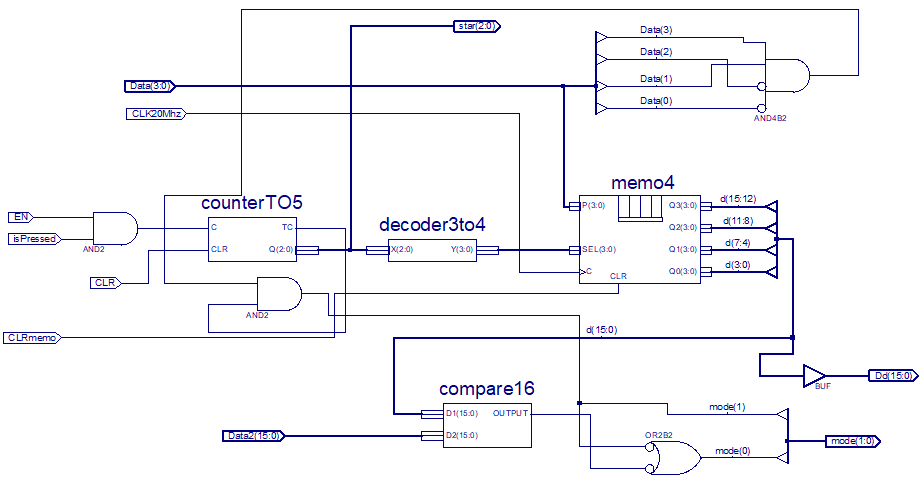
รูปที่ 5 แสดงวงจร NumBCD



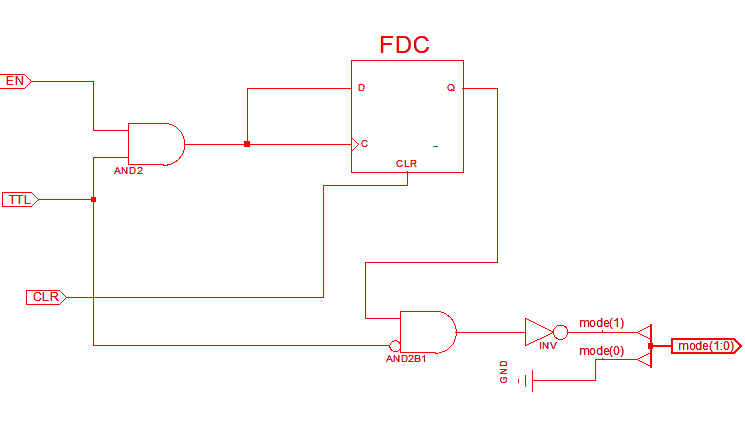
รูปที่ 6 แสดงวงจร ModeDecoder



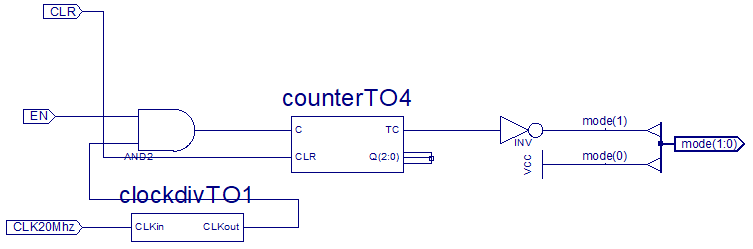
รูปที่ 7 แสดงวงจร Scan



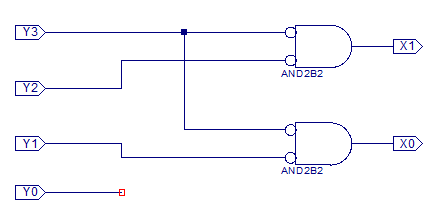
รูปที่ 8 แสดงวงจร Enterpass



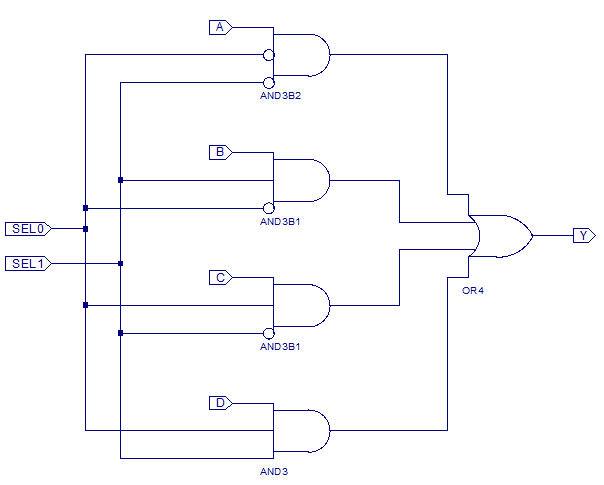
รูปที่ 9 Unlock



รูปที่ 10 แสดงวงจร Incorrect



รูปที่ 11 Encoder 4 to 2



รูปที่ 12 Mux

# PlanAhead Generated physical constraints

NET "arduiData[3]" LOC = P100;

NET "arduiData[2]" LOC = P101;

NET "arduiData[1]" LOC = P102;

NET "arduiData[0]" LOC = P104;

# PlanAhead Generated physical constraints

NET "arduiStar[2]" LOC = P105;

NET "arduiStar[1]" LOC = P111;

NET "arduiStar[0]" LOC = P112;

NET "NumIN[0]" LOC = P137;

NET "NumIN[1]" LOC = P133;

NET "NumIN[2]" LOC = P131;

NET "NumIN[3]" LOC = P126;

NET "NumOUT[0]" LOC = P2;

NET "NumOUT[1]" LOC = P143;

NET "NumOUT[2]" LOC = P141;

NET "NumOUT[3]" LOC = P139;

NET "OSC" LOC = P123;

NET "arduiCLK" LOC = P99;

# mode

NET "p67" LOC = P67;

NET "p74" LOC = P74;

NET "p75" LOC = P75;

NET "p78" LOC = P78;

#modeOut

NET "modeOut1" LOC = P114;

NET "modeOut0" LOC = P115;

NET "p82" LOC = P82; # arduiclk

NET "p81" LOC = P81; # enable

NET "p84" LOC = P84; # numpad

# star

NET "s[2]" LOC = P93;

NET "s[1]" LOC = P94;

NET "s[0]" LOC = P95;

# bcd

NET "bcdebug[3]" LOC = P85;

NET "bcdebug[2]" LOC = P87;

NET "bcdebug[1]" LOC = P88;

NET "bcdebug[0]" LOC = P92;

# sseg

NET "p41" LOC = P41;

NET "p40" LOC = P40;

NET "p35" LOC = P35;

NET "p34" LOC = P34;

NET "p32" LOC = P32;

NET "p29" LOC = P29;

NET "p27" LOC = P27;

NET "p44" LOC = P44;

NET "p43" LOC = P43;

NET "p33" LOC = P33;

NET "p30" LOC = P30;

## วัสดุอุปกรณ์ที่ใช้

1. บอร์ด FPGA รุ่น SERVEYOR-6 XCS6SLX9
2. Servo Motor 1 อัน
3. Aruduino Maker Uno 2 อัน
4. Breadboard 1 อัน
5. LCD 16x2 1อัน
6. Keypad 4x4 1 อัน
7. Buzzer 1 อัน
8. กล่องไม้ 1 อัน
9. แผ่นอะคริลิค
10. Arduino RFID RC522A MFRC522 1 อัน

## ขั้นตอนการดำเนินงาน

1. ประชุมกลุ่มเพื่อหาหัวข้อชิ้นงานที่จะจัดทำ
2. จัดทำเอกสารข้อเสนอโครงการและนำส่งแก่อาจารย์
3. ออกแบบระบบของชิ้นงานแบบ Top down Design
4. เขียนโปรแกรมตามที่ได้ออกแบบไว้
5. จัดทำชิ้นงาน
6. ทดสอบ ปรับปรุงแก้ไขชิ้นงาน
7. จัดทำรายงาน วีดีทัศน์ พร้อมนำส่งแก่อาจารย์

# **บทที่ 4 ผลการดำเนินงาน**

## วิธีการใช้ชิ้นงาน

เมื่อต้องการเปิด Smart lock box จะต้องนำ การ์ด RFID นำมาแตะที่แผงอ่าน RFID ในส่วนหน้าของ Smart lock box จากนั้นป้อนรหัสไปยัง Smart lock box ผ่าน Numpad จากนั้น อุปกรณ์ Smart lock box จะปลดล็อคออก และสามารถเปิดกล่องได้ หากนำของเข้าไปในกล่องแล้ว สามารถปิดฝาของ Smart lock box เพื่อให้อุปกรณ์ เข้าสู่โหมด Lock ดังเดิม

## Source code :

Servo myservo;

SoftwareSerial sr(0, 1); // RX, TX

String state = "WAIT\_RFID";

int data[16];

int i = 0;

void setup() {

sr.begin(9600);

Serial.begin(9600);

pinMode(CLK, OUTPUT);

pinMode(D1, OUTPUT);

pinMode(D2, OUTPUT);

pinMode(D3, OUTPUT);

pinMode(D4, OUTPUT);

pinMode(STARb2, INPUT);

pinMode(STARb1, INPUT);

pinMode(STARb0, INPUT);

pinMode(MODEb1, INPUT);

pinMode(MODEb0, INPUT);

}

void loop() {

if (state == "WAIT\_RFID") {

myservo.write(0);

int count = 0;

int tmp1;

int tmp2;

while (1) {

if (sr.available()) {

if (count == 0) {

tmp1 = sr.read();

Serial.println(tmp1);

count++;

}

else if (count == 1) {

tmp2 = sr.read();

Serial.println(tmp2);

break;

}

else {

int tmpFool = sr.read();

count++;

}

}

}

int i = 0;

int arr1[8] = {0, 0, 0, 0, 0, 0, 0, 0};

int arr2[8] = {0, 0, 0, 0, 0, 0, 0, 0};

while (tmp1 > 0) {

arr1[i] = tmp1 % 2;

tmp1 = tmp1 / 2;

i++;

}

i = 0;

while (tmp2 > 0) {

arr2[i] = tmp2 % 2;

tmp2 = tmp2 / 2;

i++;

}

for (int i = 0; i < 8; i++ ) {

data[15 - i] = arr1[i];

}

for (int i = 0; i < 8; i++ ) {

data[7 - i] = arr2[i];

}

send2FPGA();

for (int i = 0; i < 16; i++) {

Serial.print(data[i]);

}

Serial.println();

state = "WAIT\_NUMPAD";

}

else if (state == "WAIT\_NUMPAD") {

myservo.write(0);

int prev[3] = {0, 0, 0};

while (1) {

int tmp[3];

tmp[0] = digitalRead(STARb2);

tmp[1] = digitalRead(STARb1);

tmp[2] = digitalRead(STARb0);

// Serial.print(tmp[0]);

// Serial.print(tmp[1]);

// Serial.println(tmp[2]);

if (tmp[0] == 0 and tmp[1] == 0 and tmp[2] == 0 and (prev[0] != tmp[0] or prev[1] != tmp[1] or prev[2] != tmp[2])) {

// sr.print('0');

Serial.println("Nothing");

}

else if (tmp[0] == 0 and tmp[1] == 0 and tmp[2] == 1 and (prev[0] != tmp[0] or prev[1] != tmp[1] or prev[2] != tmp[2])) {

sr.print('a');

Serial.println('a');

}

else if (tmp[0] == 0 and tmp[1] == 1 and tmp[2] == 0 and (prev[0] != tmp[0] or prev[1] != tmp[1] or prev[2] != tmp[2])) {

sr.print('b');

Serial.println('b');

}

else if (tmp[0] == 0 and tmp[1] == 1 and tmp[2] == 1 and (prev[0] != tmp[0] or prev[1] != tmp[1] or prev[2] != tmp[2])) {

sr.print('c');

Serial.println('c');

}

else if (tmp[0] == 1 and tmp[1] == 0 and tmp[2] == 0 and (prev[0] != tmp[0] or prev[1] != tmp[1] or prev[2] != tmp[2])) {

sr.print('d');

Serial.println('d');

}

else if (tmp[0] == 1 and tmp[1] == 0 and tmp[2] == 1 and (prev[0] != tmp[0] or prev[1] != tmp[1] or prev[2] != tmp[2])) {

while (1) {

int tmp2[2];

tmp2[0] = digitalRead(MODEb1);

tmp2[1] = digitalRead(MODEb0);

if (tmp2[0] == 1 && tmp2[1] == 0) {

state = "UNLOCK\_BOX";

sr.print('U');

Serial.println('U');

break;

}

else if (tmp2[0] == 1 && tmp2[1] == 1) {

state = "INCORRECT";

sr.print('I');

Serial.println('I');

break;

}

}

break;

}

prev[0] = tmp[0];

prev[1] = tmp[1];

prev[2] = tmp[2];

}

}

else if (state == "UNLOCK\_BOX") {

int pos = 45;

myservo.write(pos);

while (1) {

int tmp[2];

tmp[0] = digitalRead(MODEb1);

tmp[1] = digitalRead(MODEb0);

if (tmp[0] == 0 && tmp[1] == 0) {

state = "WAIT\_RFID";

sr.print("S");

break;

}

}

}

else if (state == "INCORRECT") {

myservo.write(0);

while (1) {

int tmp[2];

tmp[0] = digitalRead(MODEb1);

tmp[1] = digitalRead(MODEb0);

if (tmp[0] == 0 && tmp[1] == 0) {

state = "WAIT\_RFID";

sr.print('S');

break;

}

}

}

}

void send2FPGA() {

for (int i = 0; i < 16; i++) {

digitalWrite(D1, data[i++]);

digitalWrite(D2, data[i++]);

digitalWrite(D3, data[i++]);

digitalWrite(D4, data[i++]);

digitalWrite(CLK, HIGH);

//Serial.print("HIGH");

delayMicroseconds(1);

digitalWrite(CLK, LOW);

//Serial.print("LOW");

delayMicroseconds(1);

}

}

# **บทที่ 5 สรุปผลการดำเนินงาน**

## สรุปผลการดำเนินงาน

จากการดำเนินการทำชิ้นงานและทดสอบแล้ว พบว่า Smart lock box ที่กลุ่มข้าพเจ้าจัดทำขึ้น ไม่สามารถทำงานได้ตามที่ตั้งเป้าหมายไว้ได้จริง กล่าวคือ ทำได้เพียงอ่านข้อมูลจาก RFID และอ่านข้อมูลจาก Keypad แต่ไม่สามารถเปรียบเทียบข้อมูลทั้งสองได้อย่างถูกต้อง อีกทั้งตัวอุปกรณ์ยังไม่เสถียรเท่าที่ควร



Link video ชิ้นงาน : https://youtu.be/HEuca6TCslQ

## ปัญหา อุปสรรค และการแก้ไข

1. สายไฟมีจำนวนเยอะ และพันกันยุ่งเหยิง

วิธีแก้ จัดเรียงสายไฟให้เป็นระเบียบ มัดส่วนที่รวมกันได้

1. อุปกรณ์พัง เสียหาย

วิธีแก้ ซื้อเผื่อไว้ก่อน และก่อนใช้งานต้องตรวจสอบให้ดีว่าต่อถูกไหม

# **แหล่งอ้างอิง**

“FPGA” [ออนไลน์]. เข้าถึงได้จาก :

https://th.wikipedia.org/wiki/เอฟพีจีเอ

“RFID” [ออนไลน์]. เข้าถึงได้จาก :

https://th.wikipedia.org/wiki/อาร์เอฟไอดี

“LCD 16x2” [ออนไลน์]. เข้าถึงได้จาก :

https://www.thaieasyelec.com/article-wiki/review-product-article/how-to-use-character-lcd-display-arduino-ch1-parallel-version.html

“Keypad 4x4” [ออนไลน์]. เข้าถึงได้จาก :

http://elec2web.blogspot.com/2016/04/arduino-uno-keypad-4x4.html

“Servo motor” [ออนไลน์]. เข้าถึงได้จาก :

http://www.advance-electronic.com/blog/detail/86/th/เซอร์โวมอเตอร์-(Servo-Motor).html

“Arduino 4x4” [ออนไลน์]. เข้าถึงได้จาก :

https://thaiarduino.club/what-is-arduino/

“Maker Uno” [ออนไลน์]. เข้าถึงได้จาก :

https://www.arduitronics.com/product/2572/cytron-maker-uno