**Mini-Clock PROJECT**

Mini-Clock

**กลุ่ม ขายตรงแบบ 300 %**

**รูปภาพประกอบด้วย ในอาคาร

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

**รายละเอียด Mini-Clock**

**1.โครงงานชิ้นนี้จะเป็นการแสดงการทำงานของตัวนาฬิกา**

**โดยจะนำผลลัพธ์ไปแสดงใน LED Dot Matrix(8x32)**

**2.โดยโครงงานชิ้นนี้จะแบ่งฟังก์ชั่นการทำงานออกเป็น 9 ฟังก์ชั่น**

**3.มีลูกเล่นเพิ่มเติมได้แก่ LDR และ Buzzer**

**การทำงานหลัก Mini-Clock**

**1. Display Time function**

**2. Display Date/Month/Year function**

**3. Setup Time function**

**4. Setup Date/Month/Year function**

**5. Alarm function**

**6. Stopwatch function**

**7. Timer function**

**8. Auto intensity function**

**9. Sound system function**

**รูปภาพประกอบแต่ละฟังก์ชั่น**

**1. Display Time function แสดงเวลา**

**2. Display Date/Month/Year function แสดงวัน/เดือน/ปี**

**รูปภาพประกอบด้วย ในอาคาร

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย ในอาคาร

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

**3. Setup Time function ตั้งเวลาใหม่**

**4. Setup Date/Month/Year function ตั้งวัน/เดือน/ปี ใหม่**

**รูปภาพประกอบด้วย ในอาคาร

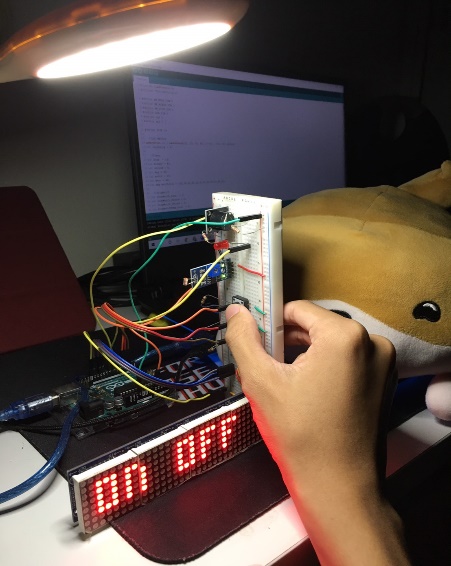
คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย ในอาคาร, ผนัง

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

**5. Alarm function นาฬิกาปลุก**

**รูปภาพประกอบด้วย ในอาคาร, ผนัง, บุคคล

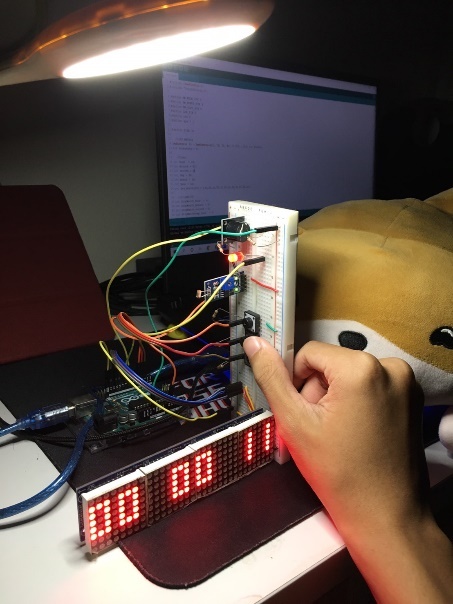
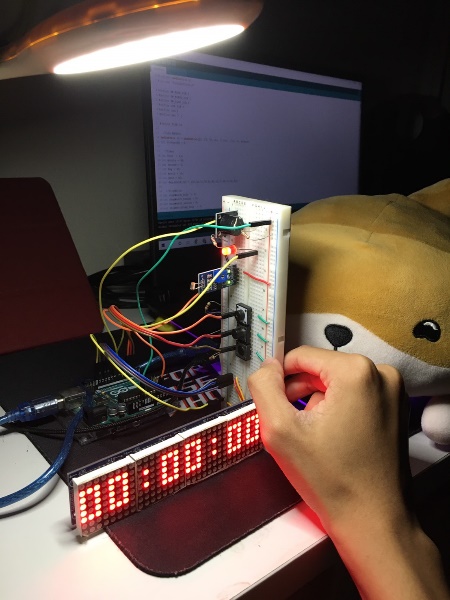
คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย บุคคล, ในอาคาร, ผนัง, ถือ

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย บุคคล, ในอาคาร, ผนัง

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย ในอาคาร

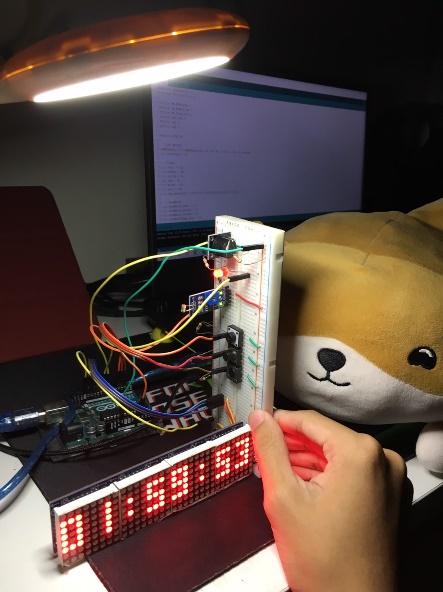
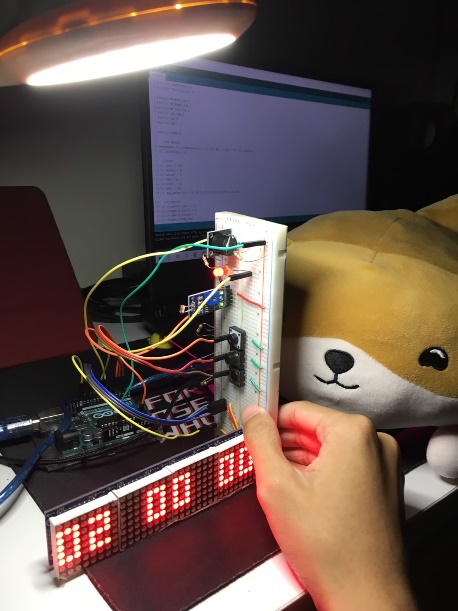
คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ ^ ภาพข้างบนเปิดปิดไฟ ^**

**6. Stopwatch function จับเวลาเดินหน้า  
รูปภาพประกอบด้วย อุปกรณ์อิเล็กทรอนิกส์

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

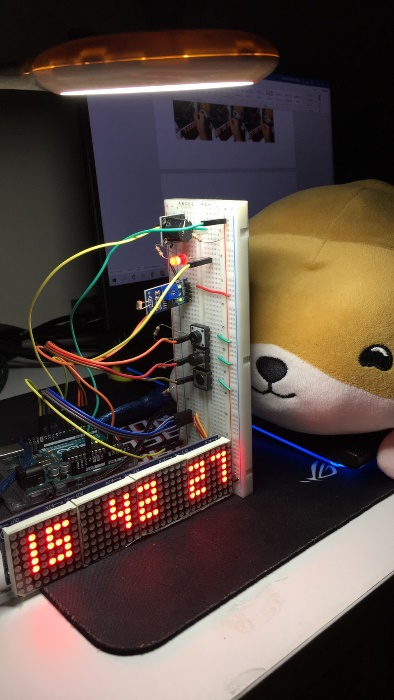
**7. Timer function จับเวลาถอยหลัง**

**รูปภาพประกอบด้วย อุปกรณ์อิเล็กทรอนิกส์

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

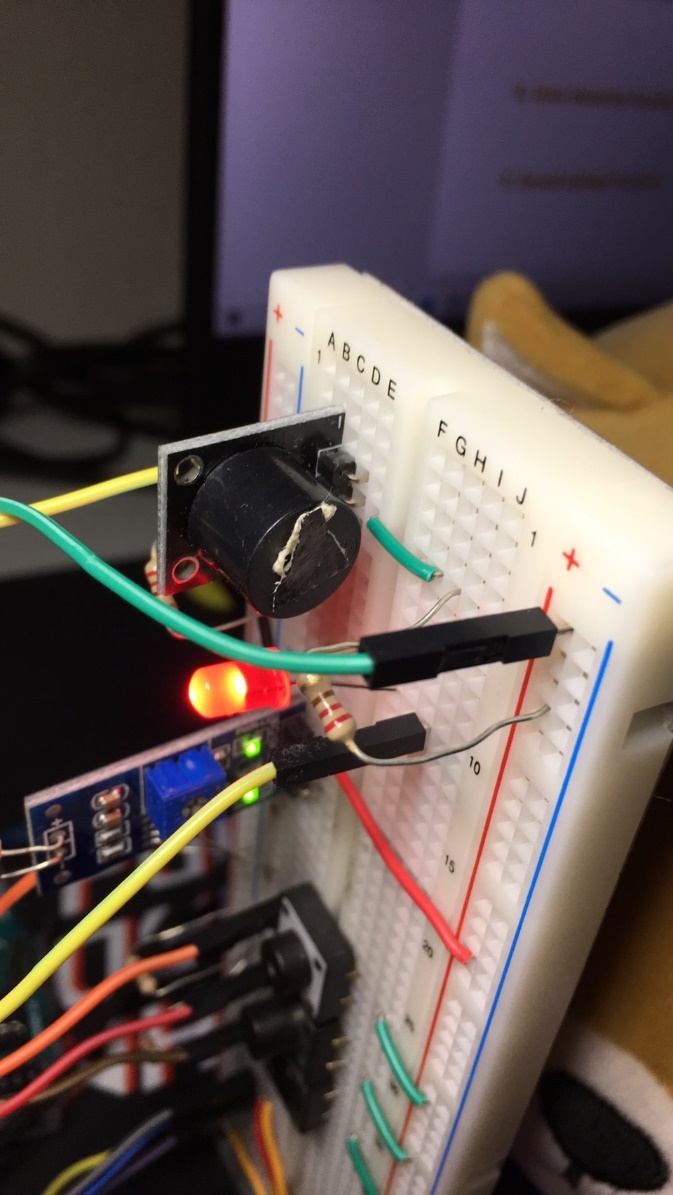
**8. Auto intensity function ปรับแสงอัตโนมัติ**

**รูปภาพประกอบด้วย อุปกรณ์อิเล็กทรอนิกส์

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย ในอาคาร, ผนัง, นั่ง, นาฬิกา

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

**9. Sound system function เสียงปุ่มกด และเสียงการเปลี่ยนโหมด**

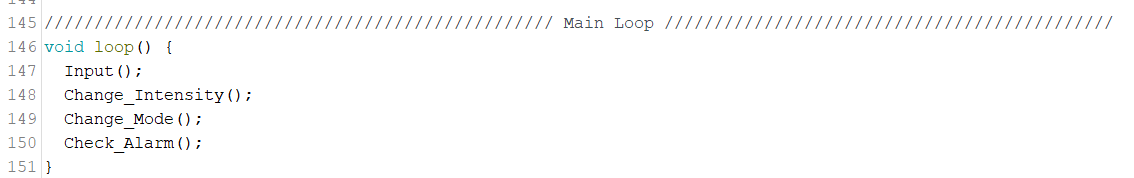
****

**อธิบายการทำงานแต่ละ Functions**

**รูปภาพประกอบด้วย ในอาคาร

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

**ฟังก์ชั่น Loop();**

****

**นี่จะเป็นการทำงานหลักทั้งหมด โดยมีฟังก์ชั่นใหญ่ๆอยู่ทั้งหมด 4 ฟังก์ชั่น**

**1. Input(); รับค่าการกดปุ่ม**

**2. Change\_Intensity(); ปรับแสง** Matrix **อัตโนมัติ**

**3. Change\_Mode(); เปลี่ยนโหมด**

**4. Check\_Alarm(); เช็คว่ามีนาฬิกาปลุกหรือไม่**

**เริ่มแรก ต้องมีการรับค่าการกดปุ่มโดยใช้ฟังก์ชั่น Input();**

**โดยปุ่มกดมีทั้งหมด 3 ปุ่ม ได้แก่ 1. ปุ่ม Mode 2. ปุ่ม Minus 3. ปุ่ม Plus**

**1.ปุ่ม Mode ใช้การ Debounce ปุ่ม แบบ กดได้ครั้งเดียว เลขจะเพิ่มค่า state\_Mode หากกดค้าง จะไม่สามารถบวก state\_Mode เพิ่มได้ ต้องปล่อยแล้วกดอีกครั้งเพื่อเพิ่มค่า**

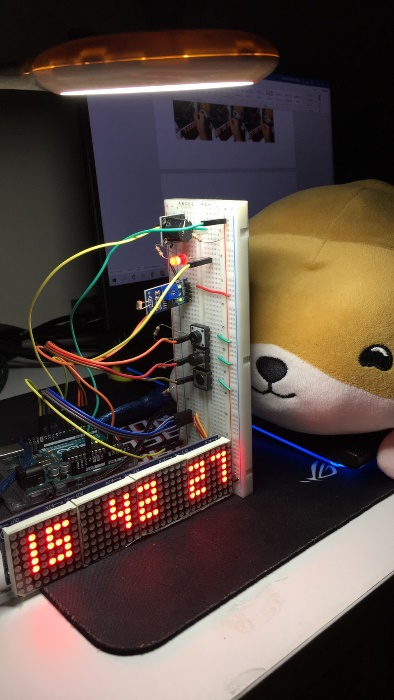
****

**2,3 ปุ่ม Minus/Plus ใช้การ Debounce ปุ่ม แบบ กดครั้งเดียว เลขจะเพิ่มค่า state\_Minus / state\_Plus หากกดค้าง จะสามารถบวกเพิ่มหรือลดลงได้ไปเรื่อยๆ**

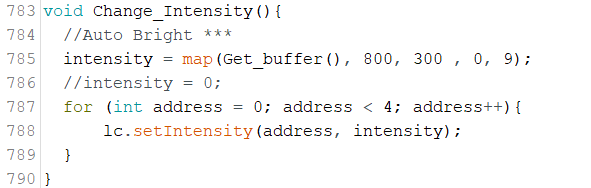
****

**ต่อมา คือการปรับความสว่างอัตโนมัติ โดยใช้ฟังก์ชั่น Change\_Intensity();**

**รูปภาพประกอบด้วย อุปกรณ์อิเล็กทรอนิกส์

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย ในอาคาร, ผนัง, นั่ง, นาฬิกา

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

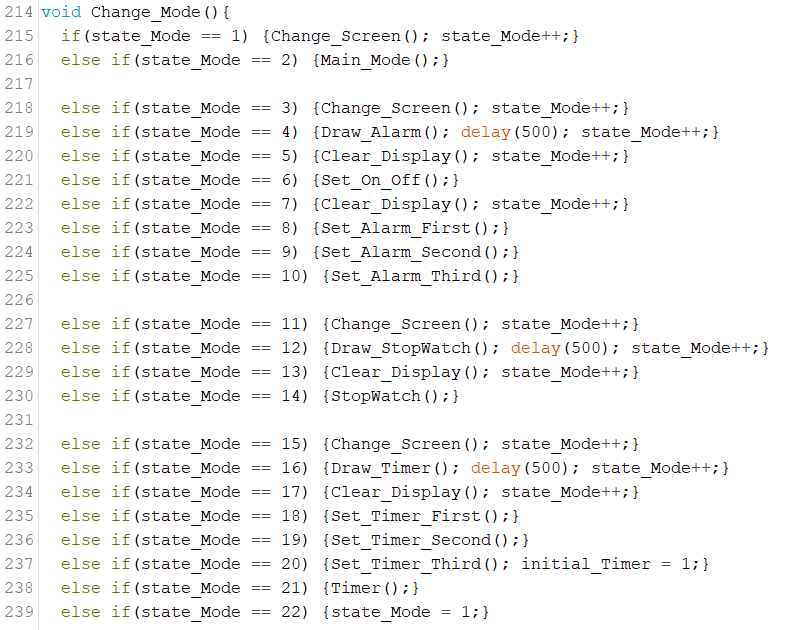
****

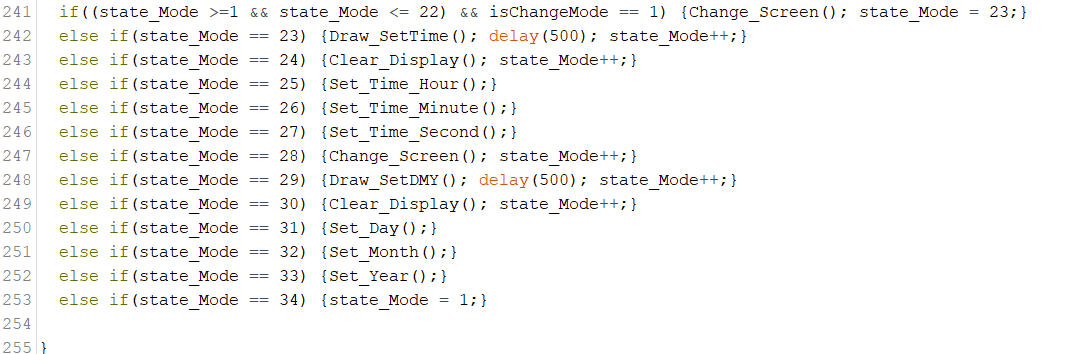
**เป็นการรับค่าความสว่างมาจาก LDR และ map ค่า ให้กลายเป็น intensity และปรับ ความสว่าง Matrix ทุกตัวโดยใช้ for\_loop**

**หลักการทำงานของมันคือ เมื่อ LDR ได้รับแสงน้อยก็จะทำให้ Matrix ไฟสว่างน้อยลง และ เมื่อ LDR ได้รับแสงมากก็จะทำให้ Matrix ไฟสว่างมากขึ้น**

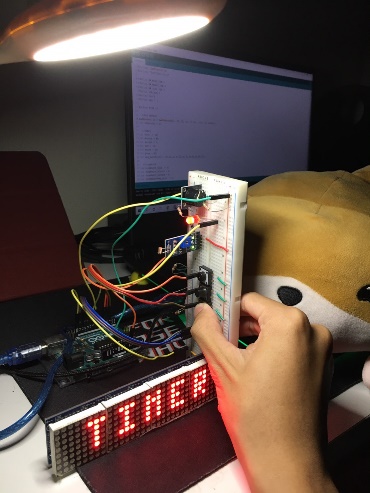
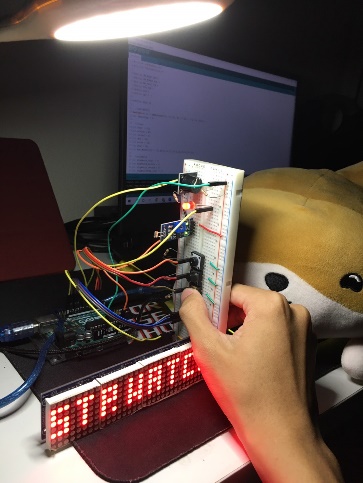
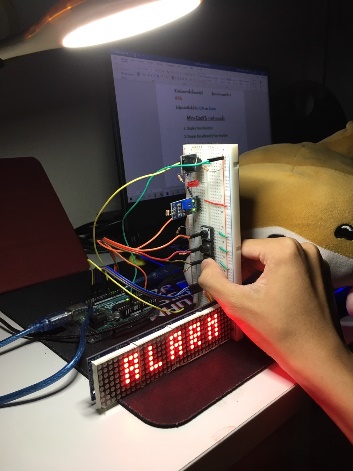
**การปรับแสงสว่างอัตโนมัติแบบนี้ นอกจากจะเป็นการประหยัดพลังงานแล้ว ยังสามารถช่วยถนอมสายตา เมื่อเรามองนาฬิกาในที่มืดได้อีกด้วย**

**ต่อมา คือการเปลี่ยนโหมดโดยใช้ฟังก์ชั่น Change\_Mode();**

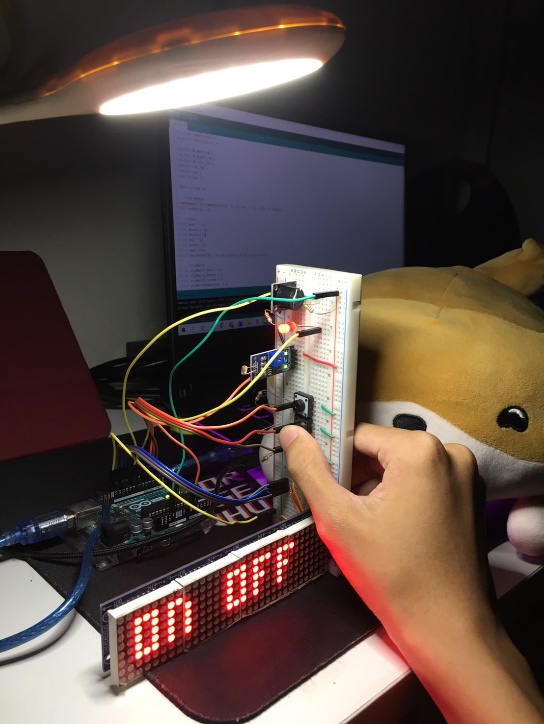
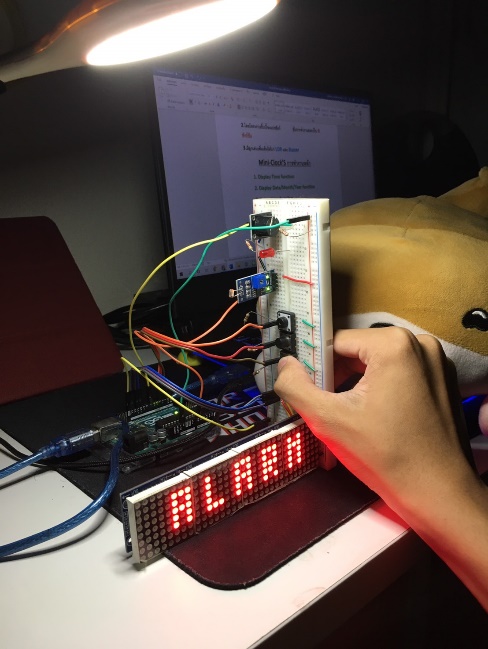
****

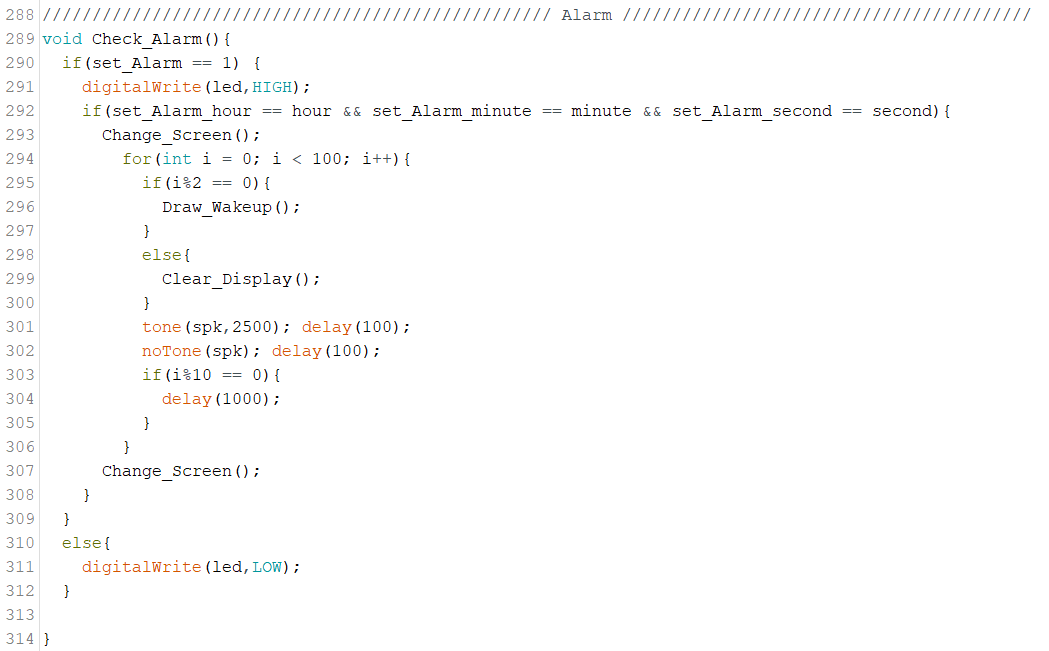
****

**โดยการเปลี่ยนโหมด ต้องกดปุ่มเพื่อเปลี่ยนโหมดไปเรื่อยๆ จากฟังก์ชั่น Input(); ที่รับค่าเข้ามาจากปุ่มกด ในที่นี้ มีทั้งหมด 34 state\_Mode เนื่องจากได้เพิ่มการ Clear\_Display(); และการ Change\_Screen(); เพื่อความสวยงามไปด้วย และยังได้เพิ่มลูกเล่นแสดงตัวอักษร ว่า ขณะนี้ กำลังอยู่ใน mode ไหน เช่น**

** เป็นต้น**

**ต่อมา คือเช็คว่ามีนาฬิกาปลุกหรือไม่ โดยใช้ฟังก์ชั่น Check\_Alarm();**

****

****

**สำหรับฟังก์นี้ จะเป็นการเช็คว่า ได้ตั้งนาฬิกาปลุกไว้หรือไม่ โดยการเช็คได้ที่ ไฟสีแดงติดหรือดับ หากไฟสีแดงติด คือตั้งนาฬิกาปลุก หากไฟสีแดงดับ คือไม่ได้ตั้งนาฬิกาปลุกไว้**

**ซึ่ง เมื่อตั้งนาฬิกาปลุกตรงกับเงื่อนไขที่ตั้ง คือเวลาปลุกตรงกับเวลาจริง จะให้มีเสียงนาฬิกาปลุกออกมา พร้อมกับ ภาพที่ขึ้นว่า “WAKE UP” แบบกระพริบ แปลว่า ตื่นได้แล้ว**

**รูปภาพประกอบด้วย ในอาคาร

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

**ฟังก์ชั่น Main\_Mode(); ประกอบไปด้วย 2 โหมดย่อย คือ**

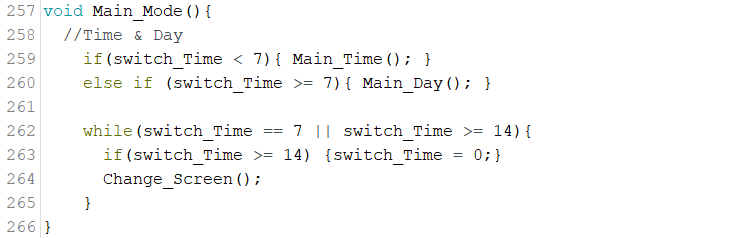
**รูปภาพประกอบด้วย ในอาคาร

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย ในอาคาร

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

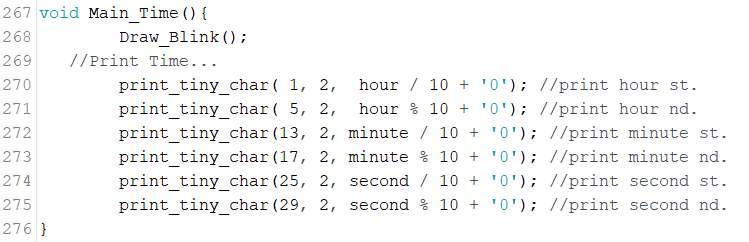
**1. Display Time function หรือ Main\_Time();**

**2. Display Date/Month/Year function หรือ Main\_Day();**

****

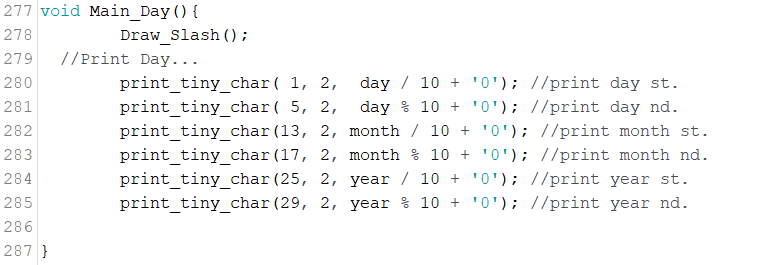
**โดย Main Functions จะมีการสลับการแสดงวันและเวลา ทุกๆ 7 วินาที สลับกันไปมา โดยใช้ ฟังก์ชั่น Change\_Screen(); ในการเปลี่ยนหน้าจอ**

1. **Display Time function หรือ Main\_Time();**

****

**แสดง เวลา เป็น hour : minute : second โดยใช้ฟังก์ชั่น print\_tiny\_char(); ในไลเบอรี่ของอาจารย์ที่แจกให้ ในการ plot จุดใน Matrixและใช้ระบบ ISR ในการเพิ่มจำนวนวินาที และฟังก์ชั่น Draw\_Blink(); แสดงการกระพริบเมื่อเวลาผ่านไป 1 วินาที**

**2. Display Date/Month/Year function หรือ Main\_Day();**

****

**แสดง วันที่ เป็น day : month : year (B.C.) โดยใช้ฟังก์ชั่น print\_tiny\_char(); ในไลเบอรี่ของอาจารย์ที่แจกให้ ในการ plot จุดใน Matrixและใช้สมการในระบบ ISR ในการคำนวณการเพิ่มขึ้นของวันเวลา และฟังก์ชั่น Draw\_Slash(); แสดงการกระพริบเมื่อเวลาผ่านไป 1 วินาที**

**หมายเหตุ: print\_tiny\_char(); / Draw\_Slash(); / Draw\_Blink(); / ISR มีอธิบายด้านหลัง**

**โหมด Set\_Time\_Mode (เป็นการเรียกรวบยอดการตั้งค่าใหม่)**

**โดยการที่เราจะเข้า Set\_Time\_Mode ได้ เราจะต้องกดปุ่ม Mode ค้างไว้ประมาณ 2 วินาที เพื่อเปลี่ยนเป็น โหมดของการตั้งค่าเวลา และวันเดือนปีใหม่ หากไม่ได้กดค้าง จะเป็นการเข้า Mode ปกติ ซึ่งก็คือ Alarm Stopwatch และ Timer(มีในหน้าต่อๆไป) นั้นเอง**

**รูปภาพประกอบด้วย ในอาคาร

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย ในอาคาร, ผนัง

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

**1. Setup Time function ประกอบไปด้วย 3 ฟังก์ชั่นย่อยคือ**

**1.1 Set\_Time\_Hour(); ตั้งค่าชั่วโมงใหม่**

**1.2 Set\_Time\_Minute(); ตั้งค่านาทีใหม่**

**1.3 Set\_Time\_Second(); ตั้งค่าวินาทีใหม่**

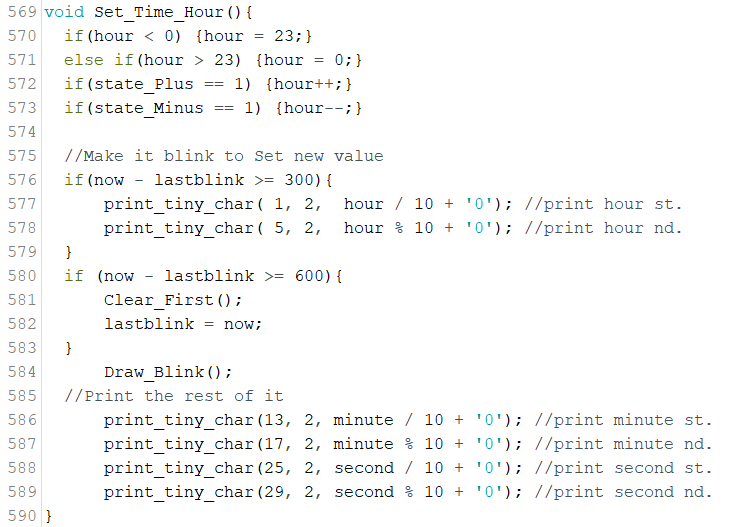
**2. Setup Date/Month/Year function ประกอบไปด้วย 3 ฟังก์ชั่นย่อยคือ**

**2.1 Set\_Day(); ตั้งค่าวันใหม่**

**2.2 Set\_Month(); ตั้งค่าเดือนใหม่**

**2.3 Set\_Year(); ตั้งค่าปีใหม่**

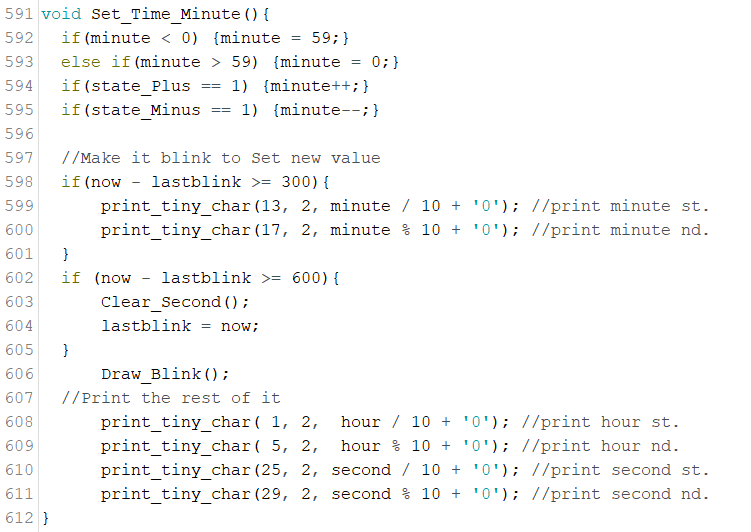
**1.1 Set\_Time\_Hour(); ตั้งค่าชั่วโมงใหม่**

****

**โดยเราจะแสดงค่าเวลาจริง และกระพริบหลัก hour เพื่อให้รู้ว่า เรากำลังเปลี่ยนหลักอะไรซึ่งเราสามารถ กดปุ่ม บวกหรือลบ เพื่อนเพิ่มค่า hour ของเราได้ โดยเมื่อกดปุ่มบวกถึง 23 ถ้ากดปุ่มบวกอีกครั้งจะกลับไปเป็น 0 ใหม่ เช่นเดียวกันกับปุ่มลบ หากกดปุ่มลบ เมื่อเลขเป็น 0 จะทำให้กลายเป็น 23 ใหม่อีกรอบ ตามความจริงของการนับชั่วโมง**

**และสร้างลูกเล่นโดยการวาดจุด ให้เหมือนกับจุดการกระพริบนาฬิกาหลักของเรา โดยใช้ฟังก์ชั่น Draw\_Blink(); เฉยๆ โดยไม่ต้องมีการกระพริบใดๆ**

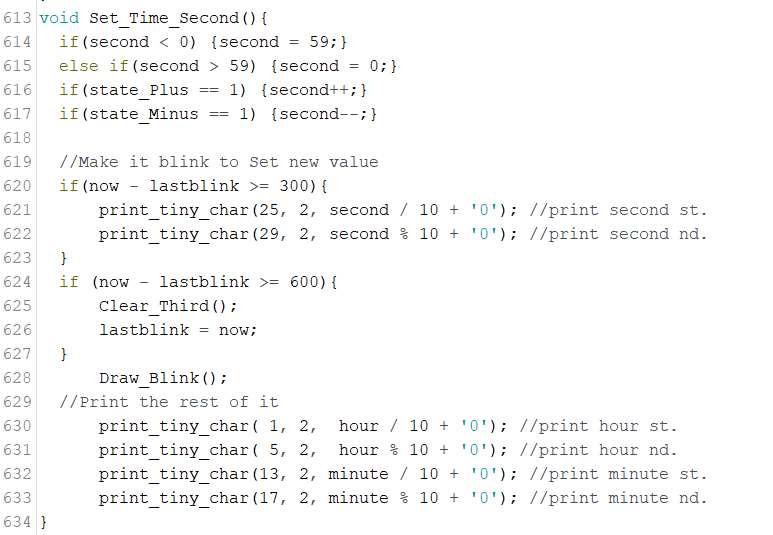
**1.2 Set\_Time\_Minute(); ตั้งค่านาทีใหม่**

****

**เช่นเดียวกับ Set\_Time\_Hour(); โดยเราจะแสดงค่าเวลาจริง และกระพริบหลัก minute เพื่อให้รู้ว่า เรากำลังเปลี่ยนหลักอะไรซึ่งเราสามารถ กดปุ่ม บวกหรือลบ เพื่อนเพิ่มค่า minute ของเราได้ โดยเมื่อกดปุ่มบวกถึง 59 ถ้ากดปุ่มบวกอีกครั้งจะกลับไปเป็น 0 ใหม่ เช่นเดียวกันกับปุ่มลบ หากกดปุ่มลบ เมื่อเลขเป็น 0 จะทำให้กลายเป็น 59 ใหม่อีกรอบ ตามความจริงของการนับนาทีของนาฬิกา**

**และสร้างลูกเล่นโดยการวาดจุด ให้เหมือนกับจุดการกระพริบนาฬิกาหลักของเรา โดยใช้ฟังก์ชั่น Draw\_Blink(); เฉยๆ โดยไม่ต้องมีการกระพริบใดๆ**

**1.3 Set\_Time\_Second(); ตั้งค่าวินาทีใหม่**

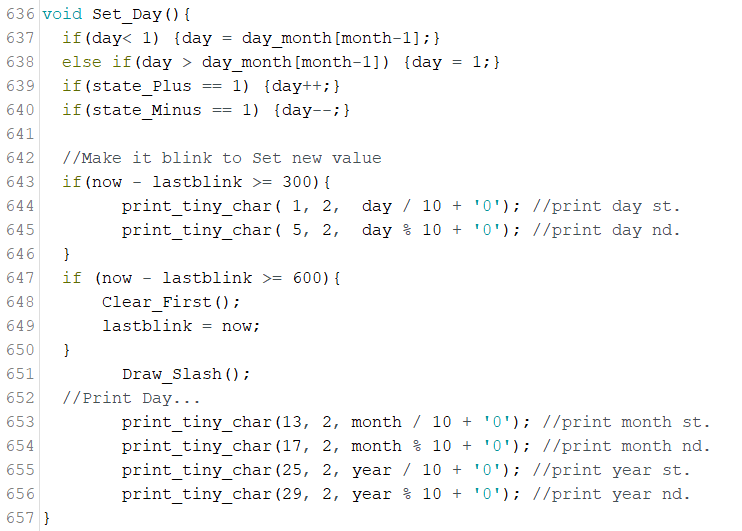
****

**เช่นเดียวกับ Set\_Time\_Minute(); โดยเราจะแสดงค่าเวลาจริง และกระพริบหลัก second เพื่อให้รู้ว่า เรากำลังเปลี่ยนหลักอะไรซึ่งเราสามารถ กดปุ่ม บวกหรือลบ เพื่อนเพิ่มค่า second ของเราได้ โดยเมื่อกดปุ่มบวกถึง 59 ถ้ากดปุ่มบวกอีกครั้งจะกลับไปเป็น 0 ใหม่ เช่นเดียวกันกับปุ่มลบ หากกดปุ่มลบ เมื่อเลขเป็น 0 จะทำให้กลายเป็น 59 ใหม่อีกรอบ ตามความจริงของการนับวินาทีของนาฬิกา**

**และสร้างลูกเล่นโดยการวาดจุด ให้เหมือนกับจุดการกระพริบนาฬิกาหลักของเรา โดยใช้ฟังก์ชั่น Draw\_Blink(); เฉยๆ โดยไม่ต้องมีการกระพริบใดๆ**

**เมื่อเราตั้งค่า ชั่วโมง นาที วินาที เสร็จ ก็จะเปลี่ยนไปเป็นการตั้งค่า วัน เดือน และ ปี**

**2.1 Set\_Day(); ตั้งค่าวันใหม่**

****

**โดยเราจะแสดงค่าวันจริง และกระพริบหลัก day เพื่อให้รู้ว่า เรากำลังเปลี่ยนหลักอะไรซึ่งเราสามารถ กดปุ่ม บวกหรือลบ เพื่อนเพิ่มค่า day ของเราได้**

**ซึ่ง day นี้จะมีการแปรผันไปตามแต่ละเดือน ซึ่งจะมีวันที่ไม่เท่ากัน จึงใช้ array ในการเก็บวันตั้งแต่เดือนแรก จนถึงเดือนสุดท้าย**

****

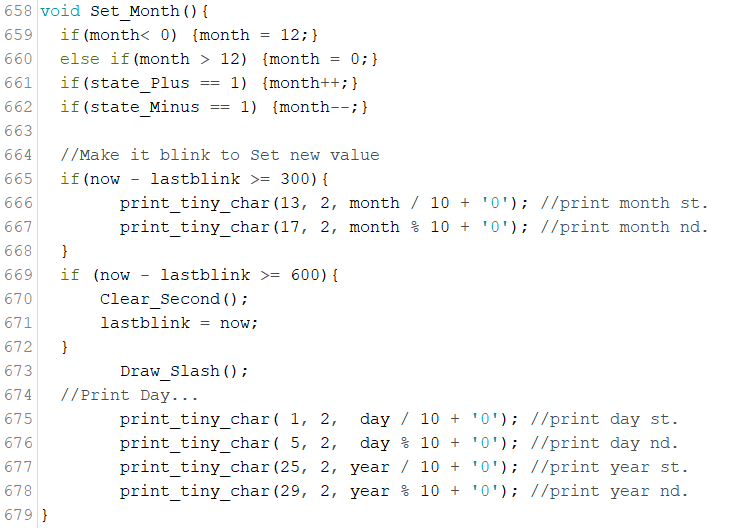
**และดักเคสต่อ เนื่องจากปี ค.ศ. ที่หาร 4 ลงตัว เดือนกุมภาพันธ์จะมี 29 วัน (แต่ในที่นี้ นาฬิกาเป็นแบบ พ.ศ. เพราะฉะนั้น จะเป็น พ.ศ.+1 ที่หาร 4 ลงตัว เดือนกุมภาพันธ์จะมี 29 วัน)**

****

**โดยเมื่อกดปุ่มบวกถึง [จำนวนวันในเดือนนั้นๆ] ถ้ากดปุ่มบวกอีกครั้งจะกลับไปเป็น 1 ใหม่ เช่นเดียวกันกับปุ่มลบ หากกดปุ่มลบ เมื่อเลขเป็น 1 จะทำให้กลายเป็น [จำนวนวันในเดือนนั้นๆ] ใหม่อีกรอบ ตามความจริงของการนับวัน**

**และสร้างลูกเล่นโดยการวาดสแลช ให้เหมือนกับสแลชที่แสดงในหน้าที่โชว์วันที่ในนาฬิกาหลักของเรา โดยใช้ฟังก์ชั่น Draw\_Slash();**

**2.2 Set\_Month(); ตั้งค่าเดือนใหม่**

****

**โดยเราจะแสดงค่าเดือนจริง และกระพริบหลัก month เพื่อให้รู้ว่า เรากำลังเปลี่ยนหลักอะไรซึ่งเราสามารถ กดปุ่ม บวกหรือลบ เพื่อนเพิ่มค่า month ของเราได้**

**โดยเมื่อกดปุ่มบวกถึง 12 ถ้ากดปุ่มบวกอีกครั้งจะกลับไปเป็น 1 ใหม่ เช่นเดียวกันกับปุ่มลบ หากกดปุ่มลบ เมื่อเลขเป็น 1 จะทำให้กลายเป็น 12 ใหม่อีกรอบ ตามความจริงของการนับเดือน**

**และสร้างลูกเล่นโดยการวาดสแลช ให้เหมือนกับสแลชที่แสดงในหน้าที่โชว์วันที่ในนาฬิกาหลักของเรา โดยใช้ฟังก์ชั่น Draw\_Slash();**

**2.3 Set\_Year(); ตั้งค่าปีใหม่**

****

**เช่นเดียวกับกับ Set\_Month(); โดยเราจะแสดงค่าปีจริง และกระพริบหลัก year เพื่อให้รู้ว่า เรากำลังเปลี่ยนหลักอะไรซึ่งเราสามารถ กดปุ่ม บวกหรือลบ เพื่อนเพิ่มค่า year ของเราได้**

**โดยเมื่อกดปุ่มบวกถึง 99 ถ้ากดปุ่มบวกอีกครั้งจะกลับไปเป็น 0 ใหม่ เช่นเดียวกันกับปุ่มลบ หากกดปุ่มลบ เมื่อเลขเป็น 0 จะทำให้กลายเป็น 99 ใหม่อีกรอบ ตามความจริงของการนับปี**

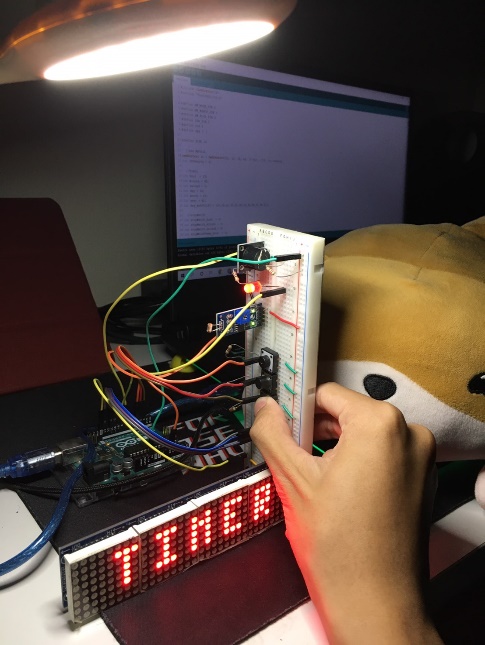
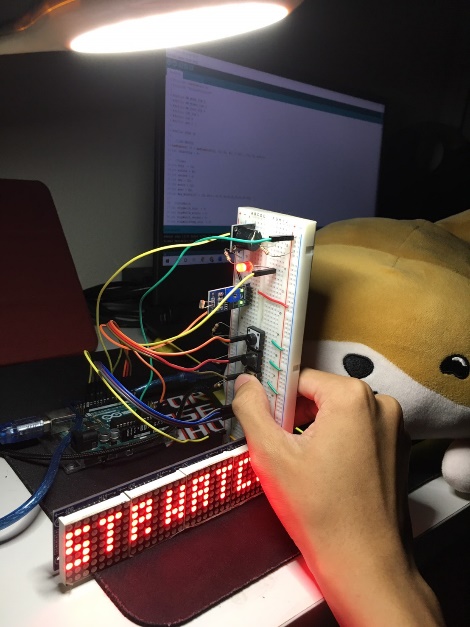
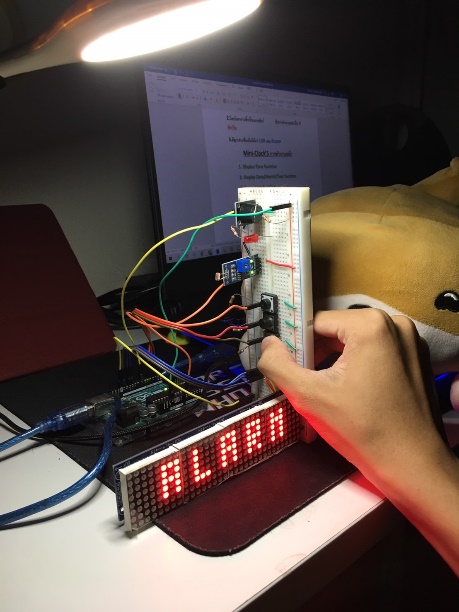
**และสร้างลูกเล่นโดยการวาดสแลช ให้เหมือนกับสแลชที่แสดงในหน้าที่โชว์วันที่ในนาฬิกาหลักของเรา โดยใช้ฟังก์ชั่น Draw\_Slash();**

**เมื่อเราตั้งค่า วันเดือนปี เสร็จ โปรแกรมก็จะทำการบันทึกค่าที่เราตั้งไว้**

**และจะรันเวลาไปเรื่อยๆตามปกติ**

**โหมด Function\_Mode (เป็นการเรียกรวบยอดการทำงานของนาฬิกา)**

**ง่ายๆโดยการกดปุ่ม Mode ก็จะสามารถเปลี่ยนจากการแสดงเวลา เป็นโหมดดังต่อไปนี้**

****

**1. Alarm function ตั้งนาฬิกาปลุก ประกอบไปด้วย 3 ฟังก์ชั่นย่อยคือ**

**1.1 Set\_Alarm\_First(); ตั้งค่าชั่วโมงปลุกใหม่**

**1.2 Set\_Alarm\_Second(); ตั้งค่านาทีปลุกใหม่**

**1.3 Set\_Alarm\_Third(); ตั้งค่าวินาทีปลุกใหม่**

**2. Stopwatch function จับเวลาเดินหน้า**

**3. Timer function จับเวลาถอยหลัง ประกอบไปด้วย 4 ฟังก์ชั่นย่อยคือ**

**3.1 Set\_Timer\_First(); ตั้งค่าจับเวลาชั่วโมง**

**3.2 Set\_Timer\_Second(); ตั้งค่าจับเวลานาที**

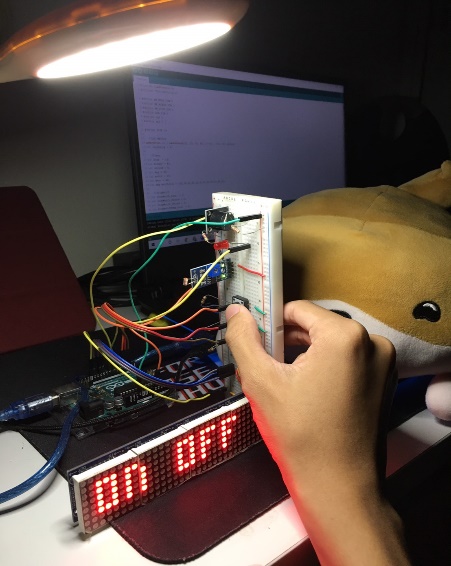
**3.3 Set\_Timer\_Third(); ตั้งค่าจับเวลาวินาที**

**3.4 Timer(); จับเวลาถอยหลัง**

**1. Alarm function ตั้งนาฬิกาปลุก**

**รูปภาพประกอบด้วย ในอาคาร, ผนัง, บุคคล

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย บุคคล, ในอาคาร, ผนัง, ถือ

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

**โดยหลักการทำงาน กดปุ่มโหมดหนึ่งครั้งเพื่อเข้าไปในโหมดนาฬิกาปลุก ซึ่งในตอนแรก จะมีให้เราเลือกว่า จะ On หรือ Off นาฬิกาปลุก ถ้า On แสดงว่า เปิดนาฬิกาปลุก ถ้า Off แสดงว่า ปิดนาฬิกาปลุก โดยจะใช้หลอดไฟ led light ในการแสดงผล เพื่อให้เราทราบ**

**เมื่อเสร็จแล้ว กดปุ่มโหมดอีกครั้งเพื่อเข้าไปตั้งค่านาฬิกาปลุก โดยใช้ฟังก์ชั่น**

**1.1 Set\_Alarm\_First(); ตั้งค่าชั่วโมงปลุกใหม่**

**1.2 Set\_Alarm\_Second(); ตั้งค่านาทีปลุกใหม่**

**1.3 Set\_Alarm\_Third(); ตั้งค่าวินาทีปลุกใหม่**

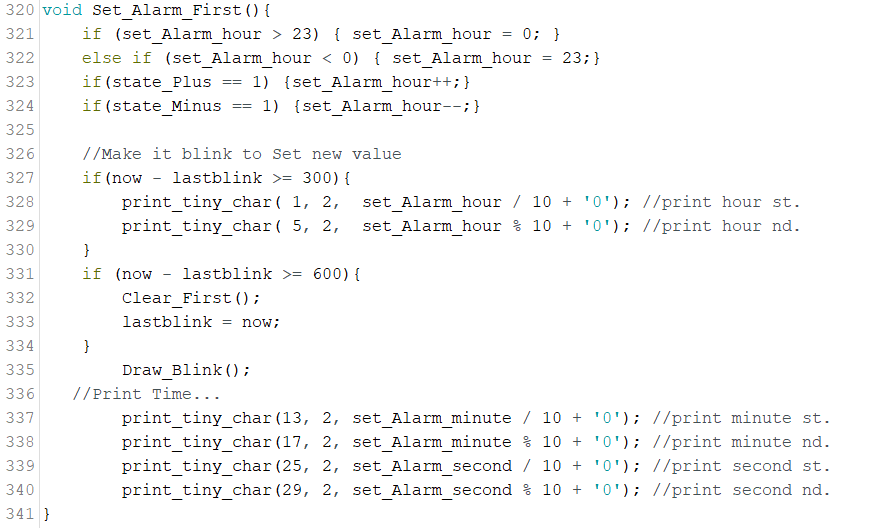
**รูปภาพประกอบด้วย บุคคล, ในอาคาร, ผนัง

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติรูปภาพประกอบด้วย ในอาคาร

คำอธิบายที่สร้างขึ้นโดยอัตโนมัติ**

**สำหรับเสียงการปลุก และลูกเล่นต่างๆสำหรับนาฬิกาปลุกได้อธิบายไว้ในฟังก์ชั่น Check\_Alarm(); เรียบร้อยแล้ว**

**1.1 Set\_Alarm\_First(); ตั้งค่าชั่วโมงปลุกใหม่**

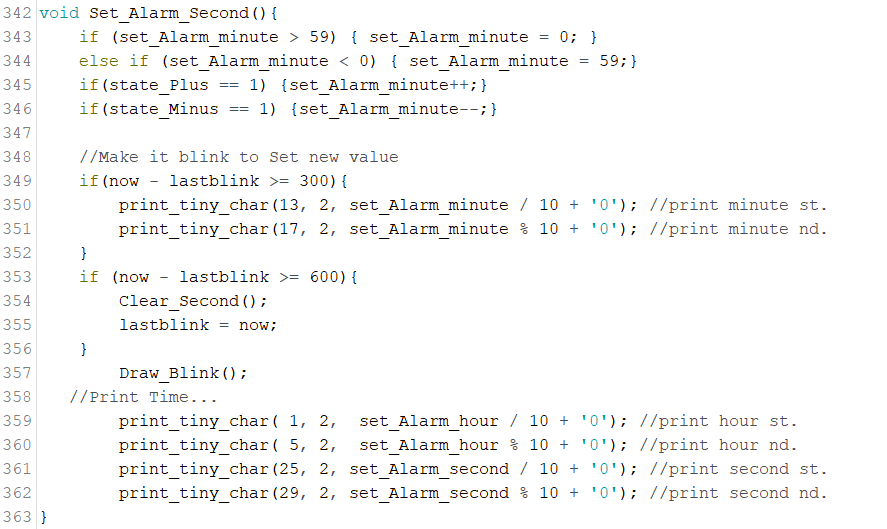
****

**หลักการทำงาน คล้ายๆกับ Set\_Time\_Hour(); ต่างกันที่ตัวแปร**

**โดยเราจะตั้งค่านาฬิกาปลุกโดยเริ่มจากหลักแรกคือ hour และกระพริบหลัก hour เพื่อให้รู้ว่า เรากำลังเปลี่ยนหลักอะไรซึ่งเราสามารถ กดปุ่ม บวกหรือลบ เพื่อนเพิ่มค่า hour ของเราได้ โดยเมื่อกดปุ่มบวกถึง 23 ถ้ากดปุ่มบวกอีกครั้งจะกลับไปเป็น 0 ใหม่ เช่นเดียวกันกับปุ่มลบ หากกดปุ่มลบ เมื่อเลขเป็น 0 จะทำให้กลายเป็น 23 ใหม่อีกรอบ ตามความจริงของการนับชั่วโมง**

**และสร้างลูกเล่นโดยการวาดจุด ให้เหมือนกับจุดการกระพริบนาฬิกาหลักของเรา โดยใช้ฟังก์ชั่น Draw\_Blink();**

**1.2 Set\_Alarm\_Second(); ตั้งค่านาทีปลุกใหม่**

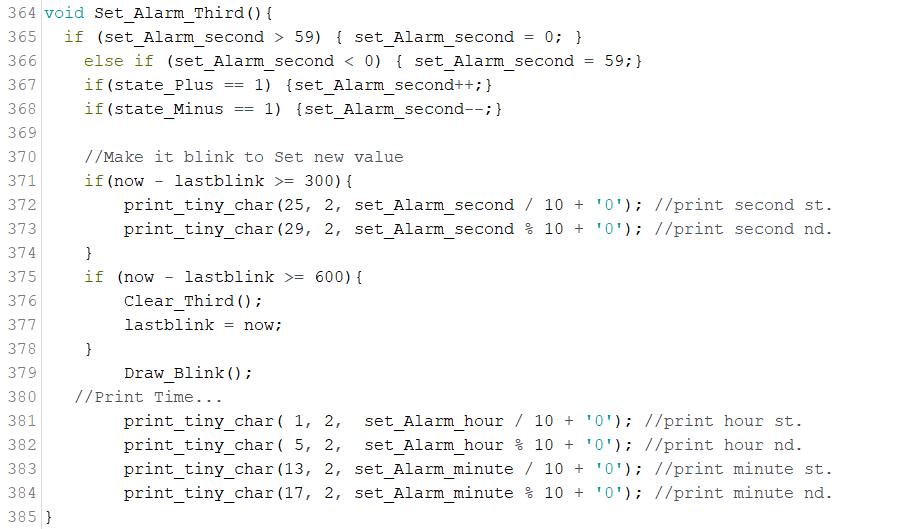
****

**หลักการทำงานเช่นเดียวกับ Set\_Alarm\_First();**

**โดยเราจะตั้งค่านาฬิกาปลุกโดยเริ่มจากหลักสองคือ minute และกระพริบหลัก minute เพื่อให้รู้ว่า เรากำลังเปลี่ยนหลักอะไรซึ่งเราสามารถ กดปุ่ม บวกหรือลบ เพื่อนเพิ่มค่า minute ของเราได้ โดยเมื่อกดปุ่มบวกถึง 59 ถ้ากดปุ่มบวกอีกครั้งจะกลับไปเป็น 0 ใหม่ เช่นเดียวกันกับปุ่มลบ หากกดปุ่มลบ เมื่อเลขเป็น 0 จะทำให้กลายเป็น 59 ใหม่อีกรอบ ตามความจริงของการนับนาทีของนาฬิกา**

**และสร้างลูกเล่นโดยการวาดจุด ให้เหมือนกับจุดการกระพริบนาฬิกาหลักของเรา โดยใช้ฟังก์ชั่น Draw\_Blink();**

**1.3 Set\_Alarm\_Third(); ตั้งค่าวินาทีปลุกใหม่**

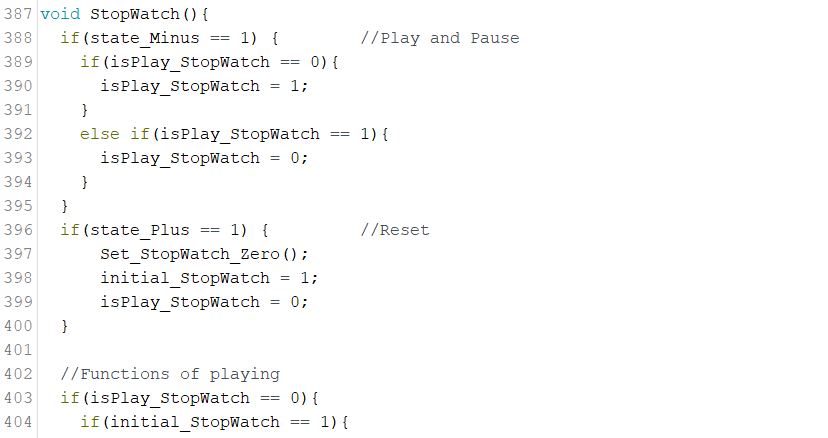
****

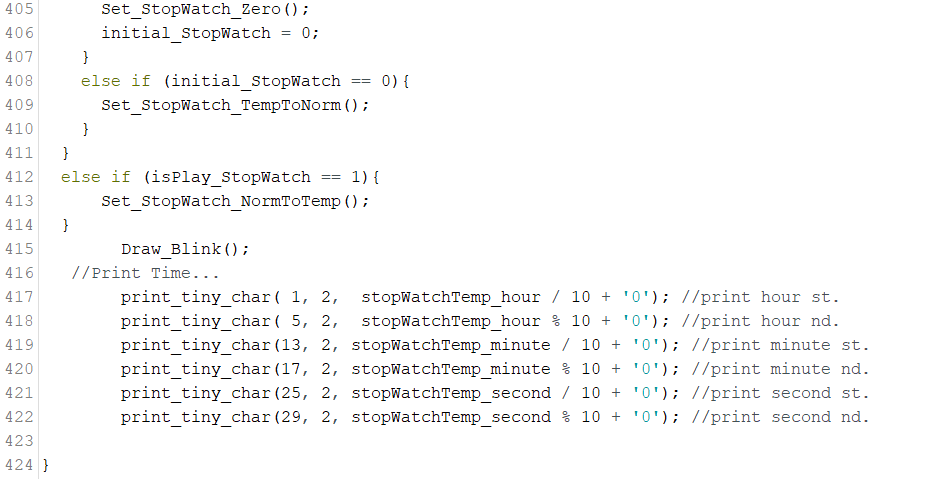
**หลักการทำงานเช่นเดียวกับ Set\_Alarm\_Second();**

**โดยเราจะตั้งค่านาฬิกาปลุกโดยเริ่มจากหลักสามคือ second และกระพริบหลัก second เพื่อให้รู้ว่า เรากำลังเปลี่ยนหลักอะไรซึ่งเราสามารถ กดปุ่ม บวกหรือลบ เพื่อนเพิ่มค่า second ของเราได้ โดยเมื่อกดปุ่มบวกถึง 59 ถ้ากดปุ่มบวกอีกครั้งจะกลับไปเป็น 0 ใหม่ เช่นเดียวกันกับปุ่มลบ หากกดปุ่มลบ เมื่อเลขเป็น 0 จะทำให้กลายเป็น 59 ใหม่อีกรอบ ตามความจริงของการนับวินาทีของนาฬิกา**

**และสร้างลูกเล่นโดยการวาดจุด ให้เหมือนกับจุดการกระพริบนาฬิกาหลักของเรา โดยใช้ฟังก์ชั่น Draw\_Blink();**

**2. Stopwatch function จับเวลาเดินหน้า**

****

****

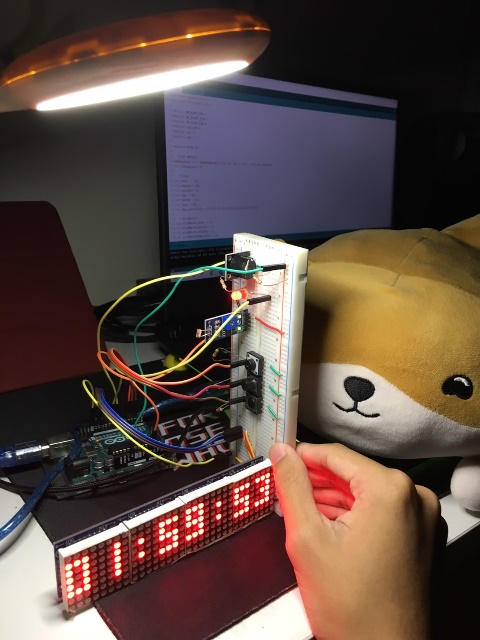
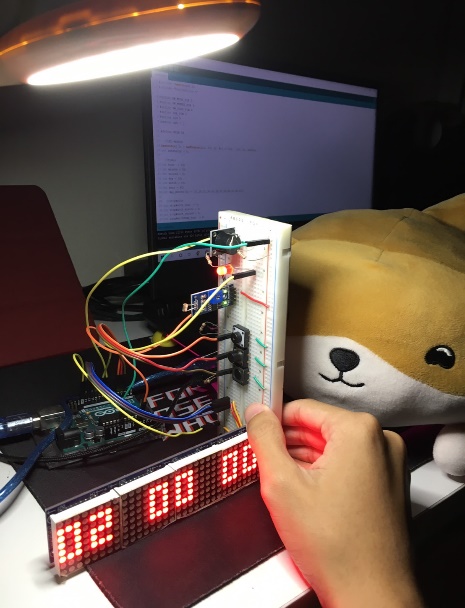
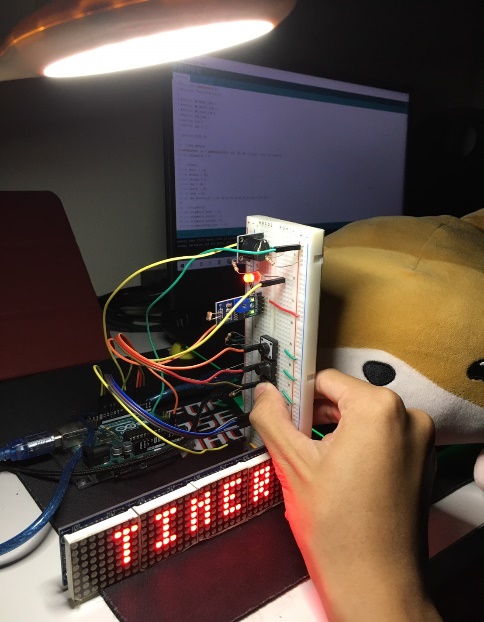
**ทำขึ้นเพื่อทำการเก็บสถิติ เช่น เด็กวิ่งในระยะทางนี้ ได้ภายในเวลาเท่าไหร่ เป็นต้น ฟังก์ชันนี้จะทำงานคล้ายๆกับฟังก์ชัน Main\_Time(); แต่จะต่างกันตรงที่ เราแค่นับเวลาเท่านั้น**

**ซึ่งเมื่อเรากดปุ่มที่สอง จะเป็นการเดินเวลา และเมื่อกดอีกทีจะเป็นการหยุดเวลา**

**เมื่อเรากดปุ่มที่สาม จะเป็นการรีเซ็ทเวลาให้เวลากลับไป 00:00**

**แต่เมื่อเรากดปุ่มแรก จะเป็นการเปลี่ยนโหมด หรือ เปลี่ยนฟังก์ชันการทำงานนั่นเอง**

**3. Timer function จับเวลาถอยหลัง**

****

**โดยหลักการทำงาน หลังจากกดปุ่มเพื่อเปลี่ยนโหมดแล้ว ก็จะมีการ set เวลาใหม่ได้แก่**

**3.** **1 Set\_Timer\_First(); ตั้งค่าจับเวลาชั่วโมง**

**3.2 Set\_Timer\_Second(); ตั้งค่าจับเวลานาที**

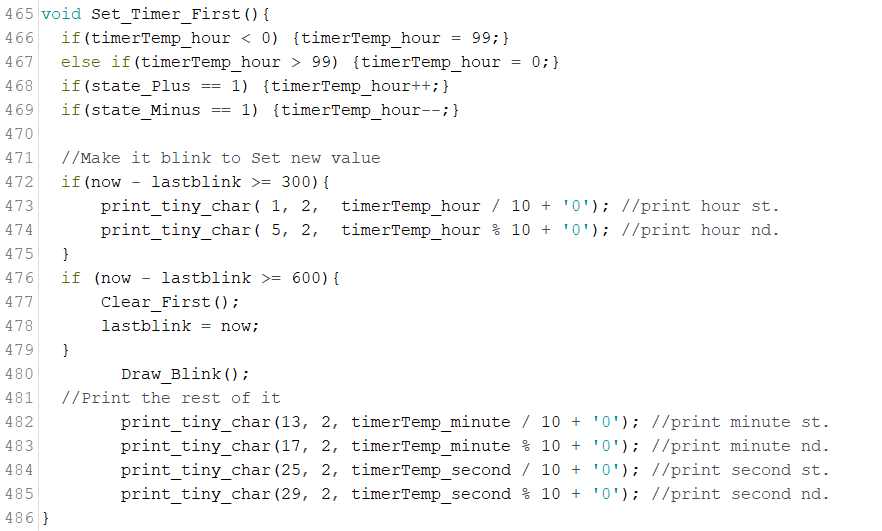
**3.3 Set\_Timer\_Third(); ตั้งค่าจับเวลาวินาที**

**การทำงานของ Timer(); นั้นทำงานคล้ายๆกับฟังก์ชัน Stopwatch(); แต่จะทำงานต่างกันตรงที่ฟังก์ชัน Timer(); จะเป็นการจับเวลาถอยหลัง เมื่อเราตั้งค่าเวลาเสร็จเรียบร้อย เวลาก็จะเริ่มเดิน และเมื่อเวลาหมด จะหยุดอยู่ที่ 00:00:00**

**3.** **1 Set\_Timer\_First(); ตั้งค่าจับเวลาชั่วโมง**

**3.2 Set\_Timer\_Second(); ตั้งค่าจับเวลานาที**

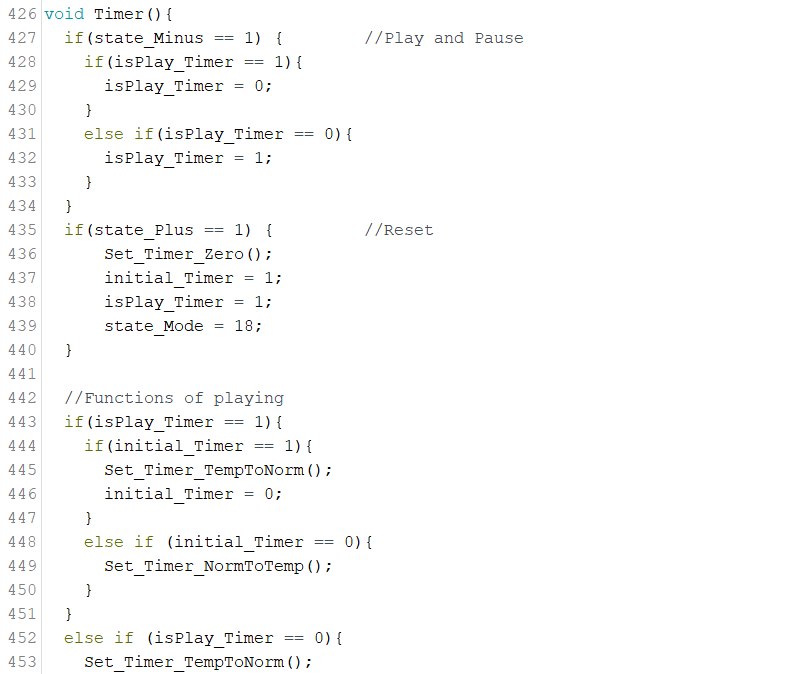
**3.3 Set\_Timer\_Third(); ตั้งค่าจับเวลาวินาที**

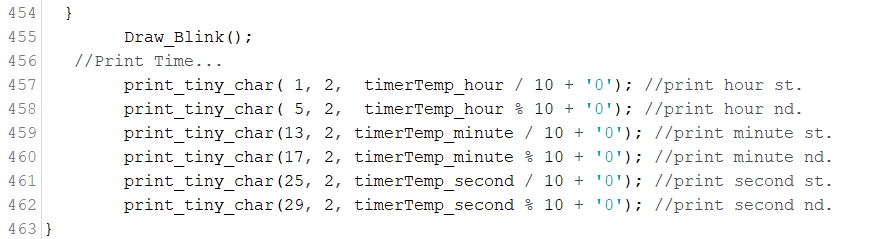
****

***ภาพตัวอย่างจาก Set\_Timer\_First();***

**โดยหลักการทำงาน เหมือนกับ Set\_Alarm\_First(); ต่างกันที่ตัวแปร**

**และทั้ง 3 ฟังก์ชั่นย่อยนี้ การทำงานเหมือนกัน ต่างกันที่ตัวแปร ก็คือไล่กระพริบจากซ้ายไปขวา เพื่อที่จะรู้ตำแหน่งการเปลี่ยนตัวเลขของแต่ละหลัก**

**3.4 Timer(); จับเวลาถอยหลัง**

****

**หลักการทำงานคล้ายๆ Stopwatch(); แต่เปลี่ยนจากเดินหน้าเป็นถอยหลังแทน ที่น่าสนใจสำหรับฟังก์ชั่นนี้คือ ต้องมีการเก็บค่าเวลาที่เรา set ไว้ในฟังกชั่นก่อนหน้า เพื่อนำมาใช้ และลดลงทุกๆ 1 second หากกดปุ่ม minus เวลาจะหยุด ถ้ากดอีกครั้ง เวลาก็จะถอยหลังต่อ ถ้ากดปุ่ม plus จะเป็นการ reset กลับไปให้เราไปตั้งค่า Timer ใหม่อีกหนึ่งรอบ หากกดปุ่ม mode จะเป็นการเปลี่ยนไปเป็นนาฬิกาปกติดังเดิม**

**อธิบายการทำงานส่วนอื่นเพิ่มเติม**

**A screenshot of a cell phone

Description automatically generated**

**ส่วนนี้จะเป็นการกำหนดค่าตัวแปรต่างๆ และใช้ lib 2ตัวก็คือ LedControl.h และ FontLEDClock.h เพื่อที่จะนำตัวโปรแกรมไปแสดงค่าบน LED Dot Matrix**

**A screenshot of text

Description automatically generated**

**ส่วนนี้จะเป็นการกำหนดค่าเวลา การหยุดเวลา และไทมเมอร์ของเวลาต่างๆ**

**A close up of text on a white background

Description automatically generated**

**ส่วนนี้จะเป็นการประกาศตัวแปร ตั้งค่า Alarm การกระพิบ บัฟเฟอร์ และกำหนดค่าปุ่ม 3 แบบ คือ เปลี่ยนโหมด เพิ่มเวลา และลดเวลา**

**A close up of text on a white background

Description automatically generated**

**ส่วนนี้จะเป็นการประกาศตัวแปร debounce ปุ่มเพื่อแก้บัคตรงที่เวลาเรากด1ครั้งจะให้ทำการเพิ่มเวลาแค่1วินาทีเท่านั้น ไม่ให้เพิ่มไปมากกว่านั้น**

**A screenshot of text

Description automatically generated**

**Set up Timer และ เสียง โดยตัว Timer เราจะใช้ prescaler เป็น 256 เนื่องจาก เราใช้บอร์ด Arduino ที่มีความถี่ขนาด 16MHz และ เสียงในส่วนนี้จะเป็นของส่วนในการเปิดเครื่อง miniclock**

**A screenshot of a social media post

Description automatically generated**

**ส่วนนี้จะเป็นส่วนของ ISR หรือ Interrupt Service Routine ซึ่งจะเป็นส่วนของโปรแกรมที่จะเกิดการทำงานแบบ Overflow ของไทมเมอร์ โดยเมื่อเวลาครบ 60 วินาทีแล้ว เราจะเปลี่ยนให้นาทีเพิ่มขึ้น 1 หน่วยเวลาและ เมื่อเวลาครบ 60 นาทีแล้ว เราจะเปลี่ยนให้ชั่วโมงเพิ่มขึ้น 1 หน่วยเวลา เป็นต้น**

**ส่วนตัวของปีนั้นเราจะแสดงเป็น พ.ศ.**

**A close up of a piece of paper

Description automatically generated**

**A close up of text on a white background

Description automatically generated**

**ส่วนนี้จะเป็นการทำงานของฟังก์ชัน Draw โดยจะทำการพิมพ์ตัวอักษรแสดงโหมดนั้นๆ**

**ได้แก่ ALARM / ON OFF / WAKE UP / STP.WATCH / TIMER / SET TIME / SET DMY**

**A screenshot of a cell phone

Description automatically generated**

**A screenshot of a cell phone

Description automatically generated A picture containing screenshot

Description automatically generated**

**เป็นฟังก์ชั่นเกี่ยวกับการแสดงบน Matrix ทั้งหมด และมีการฟังก์ชั่นการ ClearScreen ทั้งหมด และเฉพาะจุด**

**A screenshot of a social media post

Description automatically generated**

**ส่วนนี้จะเป็นการทำงานของฟังก์ชัน Get\_buffer(); เราทำบัฟเฟอร์ขึ้นมาเพื่อให้เรา map ค่าในฟังก์ชัน Intensity ได้ง่ายขึ้น เนื่องจากค่าของแสงที่ได้มาตอนแรก หรือ ค่าดิบ มันยังไม่คงที่พอที่จะนำมาในการแสดงค่าของตัว LDR**

**END OF PROJECT**