

บทที่ 2

แนวคิด ทฤษฎี และผลงานวิจัยที่เกี่ยวข้อง

งานวิจัยฉบับนี้นำเสนอต้นแบบเครื่องตรวจจับโลหะ (Metal Detector Prototype) โดยใช้บอร์ดสำเร็จรูป ET-NXP ARM KIT มาใช้ในการควบคุมการทำงานแทนบอร์ด ET-EASY168 STAMP ที่เคยนำเสนอไว้ในงานวิจัยก่อนหน้านี้ของสุทัศน์ แขกระจ่าง (2553)¹ โดยบอร์ด ET-NXP ARM KIT นี้มีการนำเอาไมโครคอนโทรลเลอร์ตระกูล ARM Cortex M3 Core เบอร์ LPC1768 ขนาด 32 บิตมาใช้งาน ซึ่งไมโครคอนโทรลเลอร์ ARM Cortex-M3 เบอร์ LPC1768 นี้มีค่าความถี่ของสัญญาณพิกาสสูงกว่าไมโครคอนโทรลเลอร์ AVR ATmega168 ถึง 5 เท่า กล่าวคือ มีค่าความถี่ของสัญญาณพิกาสในการทำงานสูงถึง 100 MHz จึงทำให้การทำงานของเครื่องมีประสิทธิภาพในการทำงานที่สูงกว่า อาทิเช่น มีความเร็วในการทำงานที่เร็วกว่ามาก มีความแม่นยำในการทำงานสูง อีกทั้งมีความละเอียดในการอ่านค่าข้อมูลที่สูง ดังนั้น ในหัวข้อที่ 2.1 จะกล่าวถึงคุณสมบัติของบอร์ด หน้าทีและการเชื่อมต่อขาพอร์ตต่างๆ ในการใช้งาน การใช้งานจอแสดงผล Graphic LCD ตัวอย่างโปรแกรมใช้งาน การใช้งานไทมเมอร์ รวมไปถึงการใช้งานอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์ LPC 1768 บนบอร์ด ET-NXP ARM KIT หัวข้อที่ 2.2 และหัวข้อที่ 2.3 จะกล่าวถึงวงจรเปิด/ปิดพัลส์ให้แก่ขดลวด (Coil Driver) และเครื่องตรวจจับโลหะที่มีใช้งานในปัจจุบันตามลำดับ ส่วนหัวข้อสุดท้ายจะกล่าวถึงผลงานวิจัยที่เกี่ยวข้อง โดยรายละเอียดของแต่ละหัวข้อย่อมีดังนี้

2.1 บอร์ดไมโครคอนโทรลเลอร์รุ่น ET-NXP ARM KIT (LPC1768)²

ET-NXP ARM KIT (LPC1768) เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล ARM Cortex M3 Core ซึ่งเลือกใช้ไมโครคอนโทรลเลอร์ 32-บิต ขนาด 100 Pin (LQFP) เบอร์ LPC1768 ของ NXP โดยระบบทรัพยากรต่างๆ ที่อยู่ในตัวของ LPC1768 ถือว่ามีความสมบูรณ์แบบมากพอสมควร เหมาะต่อการนำไป

¹ สุทัศน์ แขกระจ่าง. (2553). วงจรเปิด/ปิดพัลส์และการโปรแกรมสำหรับตรวจจับผลตอบสนองเชิงเวลาของกระแสไฟฟ้าในขดลวดที่เกิดจากโลหะโดยใช้การเหนี่ยวนำด้วยพัลส์. มหาวิทยาลัยธุรกิจบัณฑิตย์. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต.

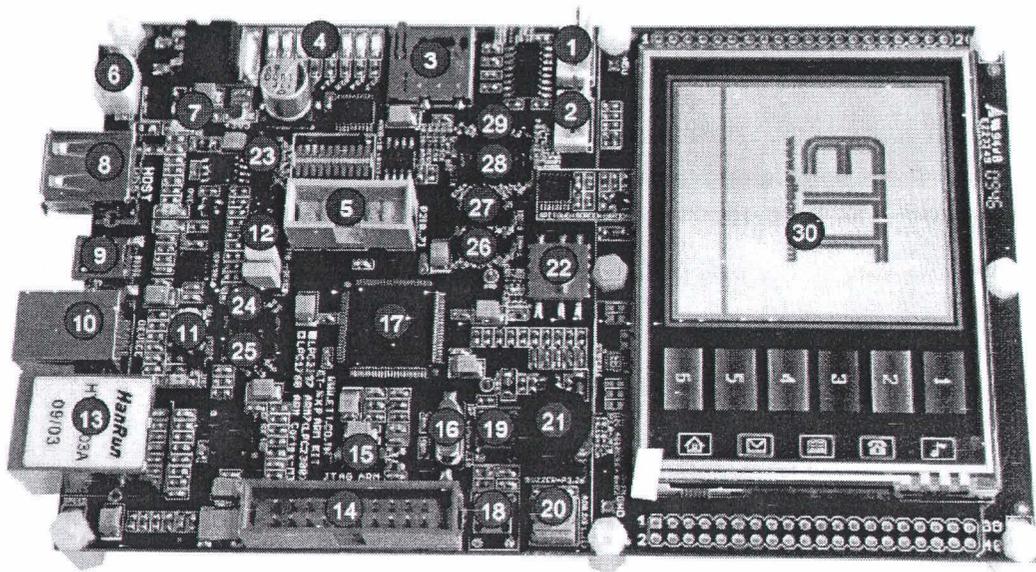
² คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์รุ่น ET-NXP ARM KIT (LPC1768). จาก

ใช้งานเพื่อศึกษาเรียนรู้ รวมไปถึงสามารถดัดแปลงแก้ไขและพัฒนาต่อยอดเพื่อสร้างเป็น Application ในรูปแบบต่างๆ ได้มากมาย เนื่องจากระบบฮาร์ดแวร์ของ LPC1768 ได้รวบรวมเอาอุปกรณ์ที่จำเป็นต่างๆ ต่อการใช้งานบรรจุไว้ภายในโครงสร้างของ MCU เพียงตัวเดียว ไม่ว่าจะเป็น ขาพอร์ต GPIO จำนวนมาก, LED สำหรับแสดงค่า Output Logic, Push Button Switch และ Joy Switch, Volume ปรับค่าแรงดัน, Mini-Speaker หรือ Buzzer, พอร์ตเชื่อมต่อ USB Device/Host/OTG, พอร์ตเชื่อมต่อหน่วยความจำแบบ SD Card, พอร์ตเชื่อมต่อ Ethernet LAN, Graphic LCD, พอร์ตเชื่อมต่อ RS-232, ADC, DAC, Timer/Counter, PWM, Capture, I²C, SPI, UART เป็นต้น

2.1.1 คุณสมบัติของบอร์ด ET-NXP ARM KIT (LPC1768)

1. ใช้ MCU ตระกูล ARM Cortex M3 เบอร์ LPC1768 ของ NXP ซึ่งเป็น MCU ขนาด 32 บิต
2. ภายใน MCU มีหน่วยความจำโปรแกรม Flash ขนาด 512 KB, Static RAM ขนาด 64 KB
3. ใช้ Crystal 12 MHz โดย MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 100 MHz เมื่อใช้งานร่วมกับ Phase-Locked Loop (PLL) ภายในตัว MCU เอง
4. มีวงจรร RTC (Real Time Clock) พร้อม XTAL ค่า 32.768 kHz และ Battery Backup
5. รองรับการโปรแกรมแบบ In-System Programming (ISP) และ In-Application Programming (IAP) ผ่านทาง On-Chip Boot-Loader Software ทางพอร์ต UART0 (RS-232)
6. มีวงจรเชื่อมต่อกับ JTAG ARM ขนาด 20 Pin มาตรฐาน เพื่อทำการ Debug แบบ Real Time
7. Power Supply ใช้แรงดันไฟฟ้า +5VDC สามารถใช้แหล่งจ่ายได้จาก 2 แหล่ง คือจากภายนอก โดยใช้ขั้วต่อแบบ 2 Pin Connector และจากขั้ว USB Device พร้อมวงจร Regulator +3V3/3A
8. มีวงจร USB Device 2.0 แบบ Full Speed ภายในตัว (USB Function มี 32 End Points)
9. มีวงจร USB Host พร้อมวงจร Over Current Protection
10. มีวงจร USB OTG โดยใช้ ISP1301 เป็น OTG (On-The-Go) Transceiver
11. มีวงจรเชื่อมต่อ Ethernet LAN 10/100Mb โดยใช้ขั้วต่อแบบ RJ-45 มาตรฐาน จำนวน 1 ช่อง
12. มีวงจรเชื่อมต่อการ์ดหน่วยความจำแบบ SD Card (Micro SD) เชื่อมต่อแบบ SPI 1 ช่อง
13. มีวงจรสื่อสาร RS-232 โดยใช้ขั้วต่อแบบ 4 พิน ตามมาตรฐานของ ETT จำนวน 2 ช่อง
14. มีวงจรเชื่อมต่อ TFT LCD Color ขนาด 320x240 Pixel (3.2 นิ้ว) พร้อม Touch Screen
15. มีวงจร Push Button Switch จำนวน 1 ชุด พร้อมสวิตช์ Reset
16. มีวงจร Joy Switch แบบ 5 ทิศทาง สำหรับใช้งาน จำนวน 1 ชุด
17. มีวงจร LED แสดงสถานะเพื่อทดลอง Output จำนวน 8 ชุด พร้อมวงจร Buffer
18. มีวงจรสร้างแรงดัน 0-3V3 โดยใช้ตัวต้านทานปรับค่าได้สำหรับทดสอบ A/D จำนวน 1 ชุด

19. มีวงจรถ้าเนิดและขับเสียง Beep โดยใช้ Mini Speaker หรือ Buzzer จำนวน 1 ชุด
20. มี 22 บิต GPIO อิสระสำหรับประยุกต์ต่างๆ เช่น D/A, I²C, I²S, CAN และ Input / Output
- Header 10 Pin IDE (P2[0..7]) สำหรับ GPIO หรือ Full-Duplex Serial UART
 - 3 Pin Header (P0[26]) สำหรับ GPIO หรือ D/A
 - 3 Pin Header (P4[29]) สำหรับ GPIO
 - 4 Pin Header (P0[19..20]) สำหรับ GPIO หรือ I²C Bus
 - 4 Pin Header (P0[0..1]) และ P0[4..5] สำหรับ GPIO หรือ CAN1 และ CAN2 Bus
 - 5 Pin Header (P0[23..25] และ P2[11..13] สำหรับ GPIO หรือ I²S-Rx และ I²S-Tx



รูปที่ 2.1 ตำแหน่งของอุปกรณ์ต่างๆ ในบอร์ด ET-NXP ARM KIT (LPC1768)

- หมายเลข 1 คือ ขั้วต่อ UART0 (RS-232) สำหรับใช้งาน และ Download Hex File ให้ CPU
- หมายเลข 2 คือ ขั้วต่อ UART2 (RS-232) สำหรับใช้งาน
- หมายเลข 3 คือ ช่องเสียบการ์ดหน่วยความจำสามารถใช้ได้กับ SD Card แบบ Micro-SD
- หมายเลข 4 คือ LED [0..7] ใช้ทดสอบ Logic Output ของ P2[0..7]
- หมายเลข 5 คือ ขั้วต่อ GPIO (P2[0..7])
- หมายเลข 6 คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ดใช้ได้กับไฟ +5VDC
- หมายเลข 7 คือ LED แสดงสถานะของ Power +VDD (+3.3 โวลต์)

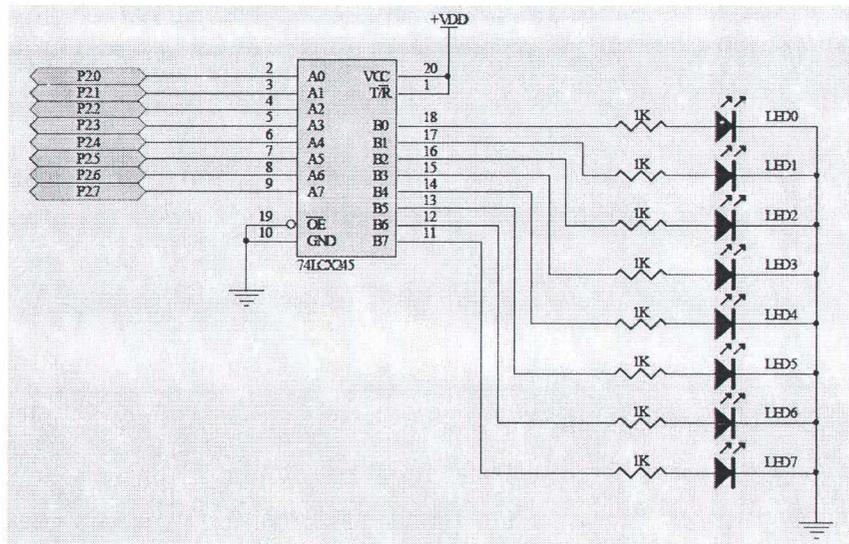
- หมายเลข 8 คือ ขั้วต่อ USB Host
- หมายเลข 9 คือ ขั้วต่อ USB OTG
- หมายเลข 10 คือ ขั้วต่อ USB Device
- หมายเลข 11 คือ Jumper (UMODE) สำหรับเลือกโหมดการ Connect ของ USB Device
- หมายเลข 12 คือ Jumper (USB) สำหรับเลือกโหมด USB ระหว่าง Device/OTG/Host
- หมายเลข 13 คือ ขั้วต่อสัญญาณ Ethernet LAN แบบ RJ45
- หมายเลข 14 คือ ขั้วต่อ JTAG ARM สำหรับ Debug แบบ Real Time
- หมายเลข 15 คือ Crystal ค่า 32.768 kHz สำหรับฐานเวลาให้ RTC ภายในตัว MCU
- หมายเลข 16 คือ Crystal ค่า 12 MHz สำหรับใช้เป็นฐานเวลาระบบให้ MCU
- หมายเลข 17 คือ MCU เบอร์ LPC1768 (100 Pin LQFP)
- หมายเลข 18 คือ SW Reset
- หมายเลข 19 คือ SW ISP Load หรือ P2.10/EINT0
- หมายเลข 20 คือ VR สำหรับปรับค่าแรงดัน 0-3.3V สำหรับทดสอบ A/D (P1[31]/AD0[5])
- หมายเลข 21 คือ Buzzer สำหรับใช้กำเนิดเสียง
- หมายเลข 22 คือ Joy Switch แบบ 5 ทิศทาง
- หมายเลข 23 คือ ขั้วต่อ CAN2 หรือ GPIO P0[4..5]
- หมายเลข 24 คือ ขั้วต่อ D/A หรือ GPIO P0.26
- หมายเลข 25 คือ ขั้วต่อ GPIO P4.29
- หมายเลข 26 คือ ขั้วต่อ CAN1 หรือ GPIO P0[0..1]
- หมายเลข 27 คือ ขั้วต่อ I²C1-Bus หรือ GPIO P0[19..20]
- หมายเลข 28 คือ ขั้วต่อ I²STx หรือ GPIO P2[11..13]
- หมายเลข 29 คือ ขั้วต่อ I²SRx หรือ GPIO P0[23..25]
- หมายเลข 30 คือ TFT LCD ขนาด 320x240 Dot พร้อม Touch Screen Sensor

2.1.2 การใช้งานวงจรขับ LED แสดงผล

LED แสดงผลของบอร์ดจะต้องวงจรแบบขับกระแส (Source Current) โดยใช้กับแหล่งจ่าย +3.3V (ขา 20) ดังแสดงในรูปที่ 2.2 ซึ่งทำงานด้วยลอจิก “1” (+3V3) และหยุดทำงานด้วยลอจิก “0” (0V) โดยควบคุมการทำงานจาก GPIO มีทั้งหมด 8 ชุด คือ P2[0..7] โดยวงจรในส่วนนี้จะใช้สำหรับทดสอบการทำงานของ Output โดยเมื่อต้องการใช้งานผู้ใช้ต้องกำหนดให้ P2[0..7] ทำหน้าที่เป็น GPIO Output Port

เสียก่อนแล้วจึงควบคุม Logic ให้กับ P2[0..7] ตามต้องการ ดังตัวอย่างในกล่องสี่เหลี่ยมข้างล่างนี้จะแสดง โปรแกรมการกำหนดค่าการใช้งาน P2[0..7] เป็น Output LED

```
// Config Pin GPIO = P2[0..7] Drive LED
LPC_PINCON->PINSEL4 &= ~(0xFFFF);           // Reset P2[0..7] = GPIO
LPC_GPIO2->FIODIR |= 0xFF;                   // P2[0..7] = Outputs
LPC_GPIO2->FIOCLR = 0xFF;                     // Turn-OFF all LED
.
.
LPC_GPIO2->FIOSET = (1<<0);                   // ON LED[0]
LPC_GPIO2->FIOCLR = (1<<0);                   // OFF LED[0]
LPC_GPIO2->FIOPIN ^= (1<<0);                  // Toggle LED[0]
```



รูปที่ 2.2 วงจรแบบขับกระแสที่ต่อกับ LED แสดงผลของบอร์ด LPC1768

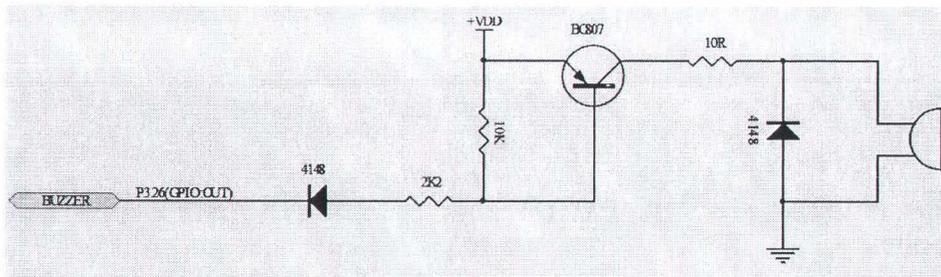
2.1.3 การใช้งานวงจรกำเนิดเสียง

วงจรถักกำเนิดเสียงจะใช้ลำโพงขนาดเล็ก (Mini Speaker) พร้อมด้วยวงจรถรานซิสเตอร์แบบ NPN สำหรับขับกระแสให้กับลำโพงโดยใช้แหล่งจ่ายขนาด +3.3V ดังแสดงในรูปที่ 2.3 ซึ่งทำงานด้วยลอจิก “1” และหยุดทำงานด้วยลอจิก “0” โดยในการทำงานนั้นต้องส่งสัญญาณลอจิกที่เป็นความถี่ต่างๆ ให้กับ

ถ้าโพงเพื่อสร้างเป็นความถี่เสียงย่านต่างๆ ตามต้องการ โดยใช้การควบคุมจาก P3[26] โดยเมื่อต้องการใช้งานผู้ใช้ต้องกำหนดให้ P3[28] ทำหน้าที่เป็น GPIO Output Port เสียก่อนแล้วจึงควบคุม Logic ให้กับ P3[28] ON/OFF เป็นความถี่ตามต้องการ ดังตัวอย่างในกล่องสี่เหลี่ยมข้างล่างนี้จะแสดงโปรแกรมการกำหนดค่าการใช้งาน P3.26 เป็น Output ขับ Buzzer

```
//Config Pin GPIO = P3[26] Drive Mini Speaker Generate Beep
LPC_PINCON->PINSEL7 &= ~(3 << 20);           // Reset P3.26 = GPIO
LPC_GPIO3->FIODIR |= (1UL<<26);              // P3[26] = Output

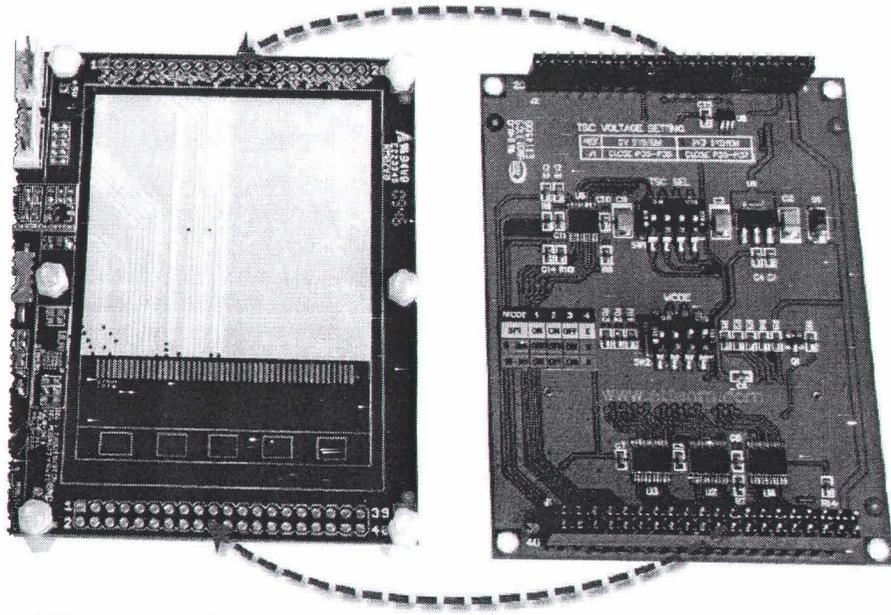
// Loop Generate Beep on Speaker(P3.26)
while(1) // Loop Continue
{
    for (i = 0; i < 500; i++) // Start Beep Pulse
    {
        LPC_GPIO3->FIOPIN ^= (1 << 26);       // Toggle P3[26]
        delay(5000);
    }
    delay(10000000); // Stop Beep Pulse
}
```



รูปที่ 2.3 วงจรกำเนิดเสียงบนบอร์ด LPC1768

2.1.4 การใช้งานจอแสดงผล Graphic LCD แบบ TFT LCD

สำหรับการเชื่อมต่อกับ Graphic LCD นั้น วงจรของบอร์ด ET-NXP ARK KIT (LPC1768) ได้รับการออกแบบให้สามารถเชื่อมต่อกับ Graphic LCD แบบ TFT LCD ขนาด 3.2 นิ้วได้ 2 รูปแบบ คือ



รูปที่ 2.4 แสดงการเชื่อมต่อกับ Graphic LCD บนบอร์ด ET-NXP ARK KIT (LPC1768)

1. ติดตั้งโมดูล LCD เข้ากับบอร์ดโดยตรงแบบถาวร โดยใช้ TFT LCD รุ่น KWH032GM02-F05 โดยใช้ในการเชื่อมต่อสัญญาณกับ TFT LCD ในโหมด SPI โดย TFT LCD รุ่นนี้จะมี Sensor ของ Touch Screen รวมอยู่ด้วย ซึ่งอุปกรณ์ที่จะใช้สำหรับอ่านค่า Sensor ของ Touch Screen วงจรของบอร์ด ET-NXP ARM KIT (LPC1768) จะออกแบบให้สามารถสามารถเลือกใช้ชิพเบอร์ STMPE811 ซึ่งใช้การเชื่อมต่อแบบ I2C หรือ อาจใช้ชิพ ADS7846 ซึ่งใช้การเชื่อมต่อแบบ SPI ก็ได้ (ขึ้นอยู่กับกาติดตั้งชิพของบอร์ดในขั้นตอนการผลิต)

2. ติดตั้งบอร์ดแสดงผลของอีทีที รุ่น ET-TFT240320TP-3.2 REV.B ซึ่งบอร์ดแสดงผลรุ่นนี้จะติดตั้ง TFT LCD รุ่น KWH032GM02-F05 พร้อมชิพ ADS7846 สำหรับอ่านค่า Touch Sensor ไว้เรียบร้อยแล้วภายในบอร์ด โดยใช้ในการติดตั้งผ่าน Connector และสามารถใส่หรือถอดออกจากบอร์ดได้โดยง่าย โดยต้องเลือกกำหนดรูปแบบการ Interface กับบอร์ดให้เป็นแบบ SPI ด้วย

รูปที่ 2.4 แสดงการเชื่อมต่อกับ Graphic LCD บนบอร์ด ET-NXP ARK KIT (LPC1768) และกล่องสี่เหลี่ยมข้างล่างนี้แสดงตัวอย่างโปรแกรมที่ใช้ในการกำหนดค่า Pin สำหรับใช้งาน GLCD และ Touch Screen

```

/* Config P1[20..24] to SSP0 For Read Touch LCD(ADS7846) */
LPC_PINCON->PINSEL3 &= ~(3UL<<10);           // Reset P1.21 Mode = GPIO
LPC_GPIO1->FIODIR |= (1UL<<21);              // P1.21 = ADS7846 CS (Output)
LPC_GPIO1->FIOPIN |= (1UL<<21);             // P1.21 = High
LPC_PINCON->PINSEL1 &= ~(3UL<<10);           // Reset P0.21 Mode = GPIO
LPC_GPIO0->FIODIR &= ~(1UL<<21);           // P0.21 = PENIRQ (Input)

//Config SSP0 Pin Connect
LPC_PINCON->PINSEL3 |= (3UL<<8);             // Select P1.20 = SCK0 (SSP0)
LPC_PINCON->PINSEL3 |= (3UL<<14);           // Select P1.23 = MISO0 (SSP0)
LPC_PINCON->PINSEL3 |= (3UL<<16);           // Select P1.24 = MOSI0 (SSP0)
LPC_SC->PCONP |= (1<<21);                   // Enable power to SSPI0 block
LPC_SC->PCLKSEL1 &= ~(3<<10);              // PCLKSP0 = CCLK/4 (18 MHz)
LPC_SC->PCLKSEL1 |= (1<<10);               // PCLKSP0 = CCLK (72 MHz)
LPC_SSP0->CPSR = 72;                        // 72 / 72 = 1MHz (max 2 MHz)
LPC_SSP0->CR0 = ( 0 << 7) |                // CPHA = 0
( 0 << 6) |                                  // CPOL = 0
( 0 << 4) |                                  // Frame format = SPI
((8-1)<< 0);                                 // Data size = 8 bits
LPC_SSP0->CR1 = ( 1 << 1);                  // Enable SSP0

/* Config P0.6,P0.7,P0.8,P0.9 to SSP1 For Control GLCD */
LPC_PINCON->PINSEL9 &= ~(3 << 24);          // Reset P4.28 Mode = GPIO
LPC_GPIO4->FIODIR |= (1UL<<28);            // Pin P4.28 = Output (BL)
LPC_GPIO4->FIOPIN &= ~(1UL<<28);           // Turn-OFF GLCD Backlight
LPC_PINCON->PINSEL0 &= ~(3UL<<12);          // Reset P0.6 Mode = GPIO
LPC_GPIO0->FIODIR |= (1 << 6);             // P0.6 = GPIO Output (CS GLCD)
LPC_GPIO0->FIOSET = (1 << 6);              // Set P0.6 = High
LPC_PINCON->PINSEL0 &= ~(3UL<<14);          // Reset P0.7 Mode = GPIO
LPC_PINCON->PINSEL0 |= (2UL<<14);          // Select P0.7 = SCK1 (SSP1)

```

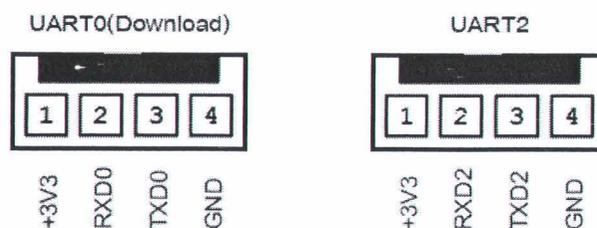


ตัวอย่างโปรแกรมที่ใช้ในการกำหนดค่า Pin สำหรับใช้งาน GLCD และ Touch Screen (ต่อ)

```
LPC_PINCON->PINSEL0 &= ~(3UL<<16); // Reset P0.8 Mode = GPIO
LPC_PINCON->PINSEL0 |= (2UL<<16); // Select P0.8 = MISO1 (SSP1)
LPC_PINCON->PINSEL0 &= ~(3UL<<18); // Reset P0.9 Mode = GPIO
LPC_PINCON->PINSEL0 |= (2UL<<18); // Select P0.9 = MOSI1 (SSP1)
LPC_SC->PCONP |= (1 << 10); // Enable power to SSP1 block
LPC_SC->PCLKSEL0 |= (2 << 20); // SSP1 clock = CCLK/2 (36 MHz)
LPC_SSP1->CPSR = 2; // Clock Rate = 18 MHz
LPC_SSP1->CR0 = (1<<7) | // CPHA = 1
(1<<6) | // CPOL = 1
(0<<4) | // Frame format = SPI
((8-1)<< 0); // Data size = 8 บิต
LPC_SSP1->CR1 = (1<<1); // Enable SSP1
```

2.1.5 การใช้งานพอร์ต RS-232

เป็นสัญญาณ RS-232 ซึ่งผ่านวงจรแปลงระดับสัญญาณ MAX3232 เรียบร้อยแล้ว โดยมีจำนวน 2 ช่อง ด้วยกันคือ UART0 และ UART2 โดยทั้ง 2 ช่องสามารถใช้เชื่อมต่อกับสัญญาณ RS-232 เพื่อรับส่งข้อมูลได้ นอกจากนี้แล้ว UART0 ยังสามารถใช้งานเป็น ISP Download สำหรับทำการ Download Hex File ให้กับ MCU ได้ด้วย โดยในกรณีนี้ต้องใช้งานร่วมกับ SW ISP Load และ SW Reset เพื่อ Reset ให้ CPU เริ่มต้นทำงานใน Boot-Loader Mode เพื่อทำการ Download Hex File ให้กับ CPU ได้ด้วย (ดูรายละเอียดเพิ่มเติมเรื่อง “การ Download Hex File ให้กับ MCU ของบอร์ด”) รูปที่ 2.5 แสดงพอร์ตเชื่อมต่อ UART0 และ UART2 บนบอร์ด ET-NXP ARK KIT (LPC1768) โดย UART-0 ใช้ขาสัญญาณจาก P0.2(TxD0) และ P0.3(RxD0) ส่วน UART-2 ใช้ขาสัญญาณจาก P0.10(TxD2) และ P0.11(RxD2)



รูปที่ 2.5 แสดงพอร์ตเชื่อมต่อ UART0 และ UART2 บนบอร์ด ET-NXP ARK KIT (LPC1768)



เนื่องจากระบบ Hardware UART ของ LPC1768/LPC2387 นั้นจะสามารถกำหนดขาสัญญาณในการเชื่อมต่อได้หลายจุด ตัวอย่างเช่น UART2 สามารถเลือกใช้ขาสัญญาณ P0[10] กับ P0[11] หรือ P2[8] กับ P2[9] ก็ได้ ซึ่งบอร์ด ET-NXP ARM KIT นั้นเลือกใช้ขาสัญญาณชุด P0[10] กับ P0[11] เป็นจุดเชื่อมต่อกับ UART2 ดังนั้น ผู้ใช้ต้องกำหนดคำสั่งในการเลือกใช้ขาสัญญาณให้ถูกต้องด้วย และข้อควรระวังอีกประการหนึ่งในการใช้งาน UART ก็คือ ค่า Default ของ UART2 จะถูกปิดการทำงานไว้ ดังนั้นผู้ใช้ต้องสั่งเปิดการทำงานของวงจร UART2 ก่อนที่จะสั่ง Initial ค่าต่างๆ ให้กับ UART ด้วย ไม่เช่นนั้นจะไม่สามารถสั่งงาน UART ได้ กล่องสี่เหลี่ยมข้างล่างนี้แสดงตัวอย่างโปรแกรมกำหนดค่า UART ในส่วนเริ่มต้นและรูปที่ 2.6 แสดงคอนเนกเตอร์ DB-9 กับ DB-25 และสายเคเบิลเพื่อเชื่อมต่อ RS-232 ระหว่าง Comport ของเครื่องคอมพิวเตอร์เข้ากับขั้วต่อ UART0 และ UART2 ของบอร์ด ET-NXP ARM KIT

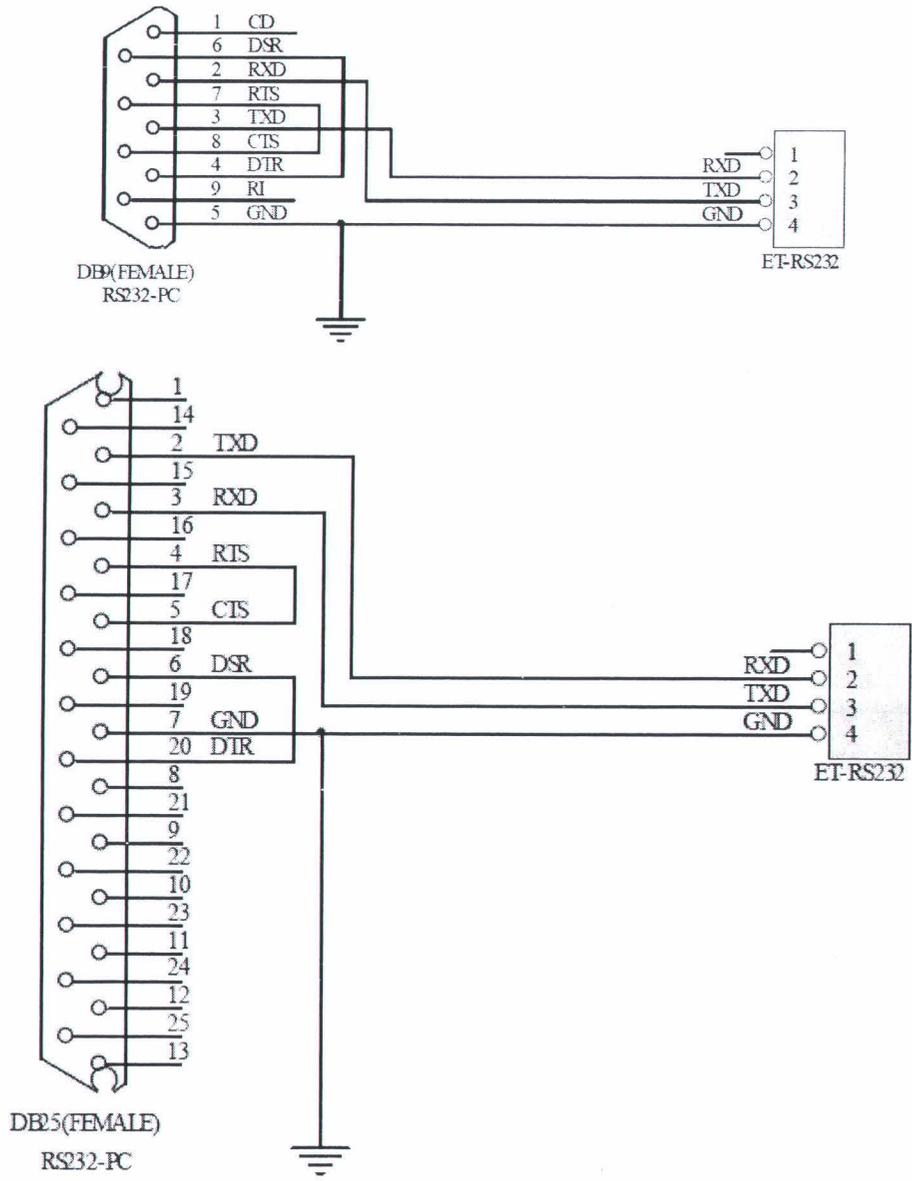
```
// Config UART0 Connect to P0[2]:P0[3]
LPC_PINCON->PINSEL0 &= ~(0x03<<4);           // Reset P0.2 = GPIO
LPC_PINCON->PINSEL0 |= (0x01<<4);             // Config P0.2 = Tx/D0
LPC_PINCON->PINSEL0 &= ~(0x03<<6);           // Reset P0.3 = GPIO
LPC_PINCON->PINSEL0 |= (0x01<<6);             // Config P0.3 = Rx/D0

// Config UART2 Connect to P0[10]:P0[11]
LPC_PINCON->PINSEL0 &= ~(0x03<<20);          // Reset P0.10 = GPIO
LPC_PINCON->PINSEL0 |= (0x01<<20);           // Config P0.10 = Tx/D2
LPC_PINCON->PINSEL0 &= ~(0x03<<22);          // Reset P0.11 = GPIO
LPC_PINCON->PINSEL0 |= (0x01<<22);           // Config P0.11 = Rx/D2
PCONP |= 0x01000000;                          // UART2 Power-ON
```

2.1.6 การใช้งานขั้วต่อ Port I/O ต่างๆ ของบอร์ด ET-NXP ARM KIT

สำหรับขั้วต่อ Port I/O ของ CPU นั้น จะจัดเรียงออกมาอย่างชัดเจนสำหรับให้ผู้ใช้เลือกต่อออกไปใช้งานตามต้องการ ซึ่งมีทั้งหมด 8 ชุด ดังแสดงในรูปที่ 2.7 โดยแต่ละขั้วต่อมีรายละเอียดดังนี้

1. ขั้วต่อ IDE 10 Pin จำนวน 2 ชุด ชุดละ 8 บิต คือ P2[0..7]
2. ขั้วต่อ DAC เป็น Header ขนาด 1x3 ใช้เป็นจุดเชื่อมต่อ P0[26] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้ทำหน้าที่เป็น D/A (Aout) ได้ โดยขา P0.26 = AOUT หรือ D/A

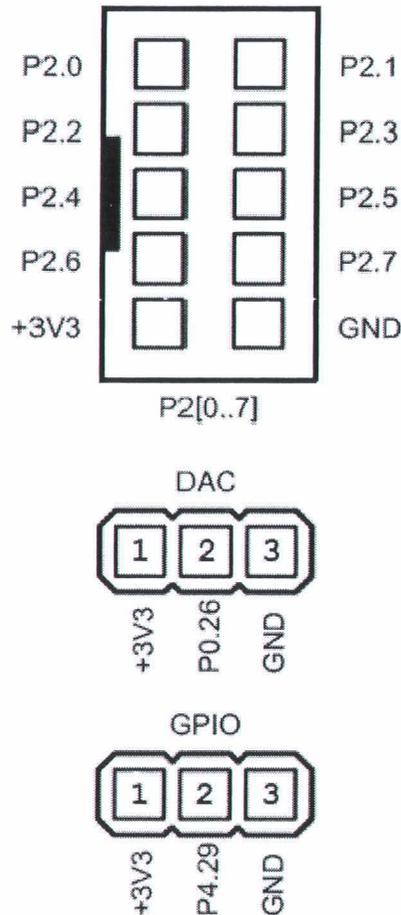


รูปที่ 2.6 แสดงคอนเนกเตอร์ DB-9 กับ DB-25 ที่เชื่อมต่อผ่านสายเคเบิลไปยังพอร์ตเชื่อมต่อ RS-232

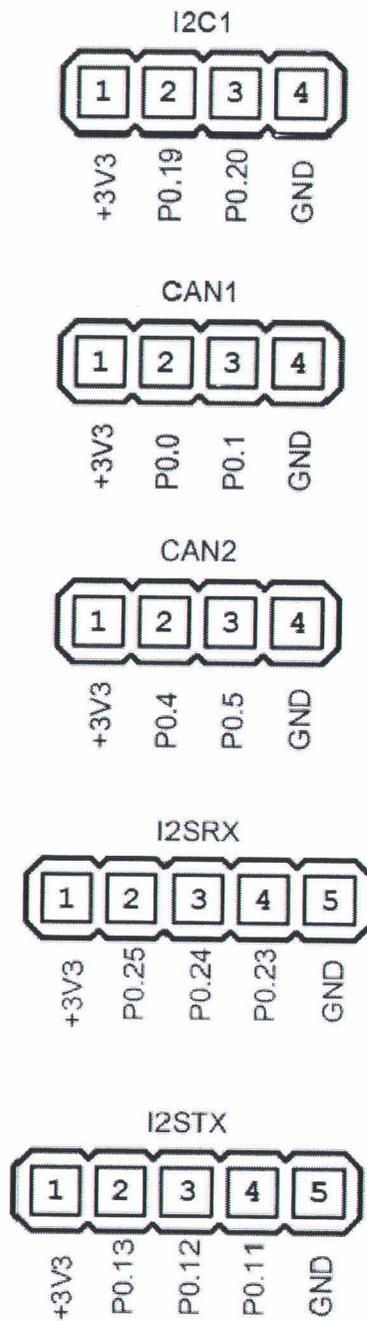
3. ขั้วต่อ GPIO เป็น Header ขนาด 1x3 ใช้เป็นจุดเชื่อมต่อ P4[29] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้ทำหน้าที่อื่นๆ ตามค่าที่คอนฟิกไว้ของ P4[29] โดยขา P4.29 = GPIO

4. ขั้วต่อ I²C-1 เป็น Header ขนาด 1x4 ใช้เป็นจุดเชื่อมต่อ P0[19..20] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้ทำหน้าที่เป็น I²C Bus ได้ โดยขา P0.19 = SDA1 และ P0.20 = SCL1

5. ขั้วต่อ CAN-1 เป็น Header ขนาด 1x4 ใช้เป็นจุดเชื่อมต่อ P0[0..1] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้เป็น Can Bus (CAN-1) ได้ โดยขา P0.0 = RD1 และ P0.1 = TD1
6. ขั้วต่อ CAN-2 เป็น Header ขนาด 1x4 ใช้เป็นจุดเชื่อมต่อ P0[4..5] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้เป็น Can Bus (CAN-2) ได้ โดยขา P0.4 = RD2 และ P0.5 = TD2
7. ขั้วต่อ I²S Rx เป็น Header ขนาด 1x5 ใช้เป็นจุดเชื่อมต่อ P0[23..25] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้เป็น I²S Rx ได้ โดยขา P0.23 = RxCLK, P0.24 = RxWS และ P0.25 = RxSDA
8. ขั้วต่อ I²S Tx เป็น Header ขนาด 1x5 ใช้เป็นจุดเชื่อมต่อ P2[11..13] ซึ่งสามารถใช้ทำหน้าที่เป็น GPIO ทั่วไป หรือใช้เป็น I²S Tx ได้ โดยขา P2.11 = TxCLK, P2.12 = TxWS และ P2.13 = TxSDA



รูปที่ 2.7 แสดงขั้วต่อ Port I/O 8 ชุด ของบอร์ด ET-NXP ARM KIT LPC1768

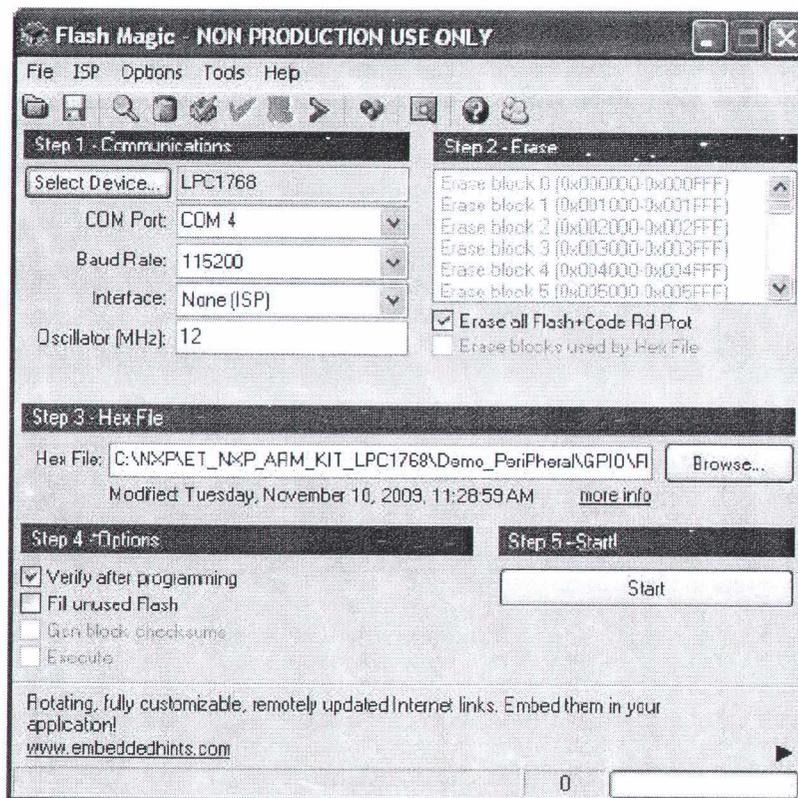


รูปที่ 2.7 แสดงขั้วต่อ Port I/O 8 ชุด ของบอร์ด ET-NXP ARM KIT LPC1768 (ต่อ)

2.1.7 การ Download Hex file ให้กับ MCU ของบอร์ดไมโครคอนโทรลเลอร์ LPC1768

การ Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ในบอร์ดนั้น จะใช้โปรแกรมชื่อ Flash Magic ของบริษัท Embedded System Academy ซึ่งจะติดต่อกับ MCU ผ่าน Serial Port ของคอมพิวเตอร์ PC โดยโปรแกรมดังกล่าวสามารถดาวน์โหลดฟรีได้ที่ www.esacademy.com โดยขั้นตอนการ Download HEX File ให้กับ MCU มีดังนี้

1. ต่อสายสัญญาณ RS232 ระหว่างพอร์ตสื่อสารอนุกรม RS232 ของ PC และบอร์ด UART0
2. จ่ายไฟเลี้ยงวงจรให้กับบอร์ด ซึ่งจะสังเกตเห็น LED PWR ติดสว่างให้เห็น
3. สั่ง Run โปรแกรม Flash Magic ซึ่งถ้าเป็น Version 5.39.1797 จะได้ผลดังรูปที่ 2.8
4. เริ่มต้นกำหนดค่าตัวเลือกต่างๆ ให้กับโปรแกรมตามต้องการ ซึ่งในกรณีนี้ใช้กับ LPC1768 ของบอร์ด ET-NXP ARM KIT (LPC1768) ของ อีทีที ให้เลือกกำหนดค่าต่างๆ ให้โปรแกรมดังนี้



รูปที่ 2.8 การใช้โปรแกรม Flash Magic Version 5.39.1797 เพื่อ Download Hex File

- 4.1 เลือก COM Port ให้ตรงกับหมายเลข COM Port ที่ใช้งานจริง (ในภาพใช้ COM4)
 - 4.2 ตั้งค่า Baud Rate อยู่ที่ระหว่าง 2400 - 115200 ซึ่งถ้าเลือกใช้ค่า Baud rate สูงๆ แล้วเกิด Error ให้ลดค่า Baud rate ให้ต่ำลง จากตัวอย่างใช้ค่า 115200
 - 4.3 กำหนด Device เป็น LPC1768
 - 4.4 กำหนด Interface เป็น None ISP
 - 4.5 กำหนดค่าคริสตอล ออสซิลเลเตอร์ ให้ตรงกับที่ใช้ในจริงภายในบอร์ด โดยกำหนดให้มีหน่วยเป็น MHz ในที่นี้ใช้ค่า 12.000 MHz ซึ่งต้องกำหนดเป็น 12
 - 4.6 ให้กดสวิตช์ ISP Load และ RESET ที่บอร์ด “ET-NXP ARM KIT” เพื่อทำการ Reset ให้ MCU ทำงานใน Boot Loader ตามขั้นตอนดังต่อไปนี้
 - 4.7 กดสวิตช์ ISP Load ค้างไว้
 - 4.8 กดสวิตช์ RESET โดยที่สวิตช์ ISP Load ยังกดค้างอยู่
 - 4.9 ปล่อยสวิตช์ RESET โดยที่สวิตช์ ISP Load ยังกดค้างอยู่
 - 4.10 ปล่อยสวิตช์ ISP Load เป็นลำดับสุดท้าย
5. เลือกรูปแบบการลบข้อมูลเป็น **“Erase all Flash + Code Rd Prot”**
 6. เลือกกำหนด Option เป็น **“Verify after programming”**
 7. ให้คลิกเมาส์ที่ **“Browse”** เพื่อทำการเลือกกำหนด HEX File ที่จะทำการสั่ง Download
 8. ให้ทำการคลิกเมาส์ที่ **“Start”** ซึ่งโปรแกรม Flash Magic จะเริ่มต้นทำการ Download ข้อมูลให้กับ MCU ทันที โดยสังเกตการทำงานที่ Status bar โดยในขั้นตอนนี้ให้รอจนกว่าการทำงานของโปรแกรมจะเสร็จสมบูรณ์
 9. เมื่อทำงานของโปรแกรมเสร็จเรียบร้อยแล้ว ให้กดสวิตช์ Reset ที่บอร์ด ซึ่ง MCU จะเริ่มต้นทำงานตามโปรแกรมที่สั่ง Download ให้ทันที

2.1.8 โมดูลพอร์ตอินพุต/เอาต์พุต (GPIO) ของ LPC1768

พอร์ตอินพุต/เอาต์พุตหรือ GPIO (General Purpose Input/Output) ภายในไมโครคอนโทรลเลอร์ LPC1768 มีความสามารถในการทำงานได้หลายโหมด ได้แก่ โหมดอินพุตลอย (Input Floating), โหมดอินพุตแบบมีพูลอัพ (Input Pull-Up), โหมดอินพุตแบบมีพูลดาวน์ (Input Pull-Down), โหมดอินพุตแอนะล็อก (Analog-Input), โหมดเอาต์พุตแบบเดรนเปิด (Output Open-Drain), โหมดเอาต์พุตแบบพุชพูล (Output Push-Pull), โหมดพุชพูลสลับฟังก์ชัน (Alternate Function Push-Pull) และโหมดเดรนเปิดสลับฟังก์ชัน (Alternate Function Open-Drain)

โมดูล GPIO จัดอยู่ในกลุ่มเพริเฟอรัล APB2 แต่ความถี่ของสัญญาณที่ออกไปยังขาพอร์ตในโหมดเอาต์พุตมีค่าสูงสุดได้ไม่เกิน 50 MHz โดยกลุ่มรีจิสเตอร์ควบคุมการทำงานของโมดูล GPIO มีดังนี้

CRL	รีจิสเตอร์ควบคุมพอร์ตไบต์ต่ำ (Port Control Register low)
CRH	รีจิสเตอร์ควบคุมพอร์ตไบต์สูง (Port Control Register High)
IDR	รีจิสเตอร์ข้อมูลอินพุต (Input Data Register)
ODR	รีจิสเตอร์ข้อมูลเอาต์พุต (Output Data Register)
BSRR	รีจิสเตอร์เซต/รีเซตบิต (Bit Set Reset Register)
BRR	รีจิสเตอร์รีเซตบิต (Bit Reset Register)
LCKR	รีจิสเตอร์ป้องกัน (Lock Register)
EVCR	รีจิสเตอร์ควบคุมอีเวนต์ (Event Control Register)
MAPR	รีจิสเตอร์สลับฟังก์ชันคีย์ของพอร์ต (Remap Debug and Register)
EXTICR	รีจิสเตอร์ตั้งค่าการอินเทอร์รัปต์ภายนอก (EXTI 0 to 15 Configuration Register)

นอกจากขาพอร์ตของ LPC1768 จะสามารถทำหน้าที่เป็นพอร์ตอินพุตเอาต์พุตใช้งานทั่วไปแล้ว ในบางตำแหน่งขายังทำหน้าที่ร่วมกับเพริเฟอรัลต่างๆ เช่น เป็นขารับ/ส่งข้อมูลของโมดูล USART, I²C, SPI, CAN หรือไอทีเมอร์ เป็นต้น รวมทั้งการสนับสนุนการสลับตำแหน่งขาพอร์ตหรือที่เรียกว่า Remap ซึ่งทำให้มีความยืดหยุ่นในการใช้งานมากขึ้น โดยความสามารถในส่วนนี้จะมีโมดูล AFIO (Alternate-Function Input/Output) เป็นตัวควบคุม

2.1.8.1 การทำงานของขาพอร์ตโหมดอินพุต

เมื่อขาพอร์ตได้รับการกำหนดให้ทำงานเป็นอินพุต กระบวนการทำงานต่อไปนี้จะเกิดขึ้น

- วงจรบัฟเฟอร์ทางเอาต์พุตถูกคิเสเบิล (ไม่ทำงาน) และวงจรอินพุตชนิดตริกเกอร์ทำงาน
- จะเกิดการพูลอัพหรือพูลดาวน์ด้วยค่าของตัวต้านทานหรือปล่อยลอยขึ้นอยู่กับโหมดของอินพุตที่เลือก
- ข้อมูลที่ปรากฏขึ้นที่ขาพอร์ตจะได้รับการสุ่มเข้าไปเก็บไว้ในรีจิสเตอร์ข้อมูลอินพุตในทุกๆ ไชเกิลของสัญญาณนาฬิกาของบัส APB2
- ทำการอ่านข้อมูลออกจากรีจิสเตอร์ข้อมูลอินพุต

2.1.8.2 การทำงานของขาพอร์ตโหมดเอาต์พุต

เมื่อขาพอร์ตได้รับการกำหนดการทำงานเป็นเอาต์พุต กระบวนการทำงานต่อไปนี้จะเกิดขึ้น

- วงจรบัฟเฟอร์ทางเอาต์พุตได้รับการเอินเอเบิลให้ทำงาน

- ถ้าหากเลือกวงจรเอาต์พุตเป็นแบบเดรนเปิด จะทำให้ PMOS ไม่ทำงาน ขาพอร์ตเอาต์พุตจะลอย ดังนั้นในการใช้จริงจะต้องมีการต่อต้านโหลดกับไฟเลี้ยงเข้าที่ขาพอร์ต
 - ถ้าหากเลือกวงจรเอาต์พุตเป็นแบบพุชพูล จะทำให้ PMOS และ NMOS สลับกันทำงานตามลอจิกที่ได้รับ
- วงจรอินพุตชmitt์ทริกเกอร์ทำงาน และทำคิสเอเบิลการพุลอัพและพุลดาวน์
 - ข้อมูลที่ปรากฏขึ้นที่ขาพอร์ตจะได้รับการสุ่มเข้าไปเก็บไว้ในรีจิสเตอร์ข้อมูลอินพุตในทุกๆ ไชเกิลของสัญญาณนาฬิกาของบัส APB2
 - ถ้าเลือกให้ทำงานในโหมดเอาต์พุตแบบเดรนเปิด ต้องอ่านข้อมูลออกจากรีจิสเตอร์ข้อมูลอินพุต
 - ถ้าเลือกให้ทำงานในโหมดเอาต์พุตแบบพุชพูล ต้องอ่านข้อมูลออกจากรีจิสเตอร์ข้อมูลเอาต์พุต

2.1.8.3 การทำงานของขาพอร์ตโหมดสลับฟังก์ชัน

เมื่อขาพอร์ตได้รับการกำหนดให้มีการสลับฟังก์ชัน กระบวนการทำงานต่อไปนี้จะเกิดขึ้น

- วงจรบัฟเฟอร์ทางเอาต์พุตถูกเปิด และสามารถกำหนดให้เป็นเดรนเปิดหรือพุชพูล
- วงจรบัฟเฟอร์ทางเอาต์พุตถูกขับด้วยสัญญาณจากเพอร์ิเฟอรัลที่ถูกกำหนดให้สลับฟังก์ชันไปทำงาน
- วงจรอินพุตชmitt์ทริกเกอร์ทำงาน
- คิสเอเบิลการพุลอัพและพุลดาวน์
- ข้อมูลที่ปรากฏขึ้นที่ขาพอร์ตจะได้รับการสุ่มเข้าไปเก็บไว้ในรีจิสเตอร์ข้อมูลอินพุตในทุกๆ ไชเกิลของสัญญาณนาฬิกาของบัส APB2
- ถ้าเลือกทำงานในโหมดเอาต์พุตแบบเดรนเปิด ต้องอ่านข้อมูลออกจากรีจิสเตอร์ข้อมูลอินพุต
- ถ้าเลือกทำงานในโหมดเอาต์พุตแบบพุชพูล ต้องอ่านข้อมูลออกจากรีจิสเตอร์ข้อมูลเอาต์พุต

2.1.9 โมดูลไทมเมอร์ (Timer) ของ LPC1768

2.1.9.1 คุณสมบัติของไทมเมอร์ควบคุมพิเศษ (Advanced Control Timer) TIM1

- ประกอบด้วยตัวนับขนาด 16 บิตที่สามารถกำหนดให้นับขึ้น, ลง, นับขึ้นและลง รวมถึงสามารถรีโหลดค่าการนับอัตโนมัติ
- มีพรีสเกลเลอร์ (Prescaler) 16 บิตแบบโปรแกรมได้ โดยสามารถกำหนดการหารความถี่สัญญาณนาฬิกาได้ตั้งแต่ 1 ถึง 65,535

- มีขาต่อใช้งาน 4 ช่อง ที่สามารถรองรับการทำงานในโหมดต่างๆต่อไปนี้ได้อย่างอิสระต่อกัน ประกอบด้วย โหมดตรวจจับสัญญาณอินพุต (Input Capture), โหมดเปรียบเทียบข้อมูล (Output Compare), โหมดกำเนิดสัญญาณ PWM (PWM Generation), โหมดสร้างสัญญาณพัลส์เดี่ยว (One Pulse Mode Output) และโหมดเอาต์พุตคอมพลีเมนต์ารีเพื่อสร้างสัญญาณ PWM แบบพิเศษสำหรับควบคุมมอเตอร์ที่สามารถกำหนดค่าเวลาวิกฤติได้ (Complementary Output with Programmable Dead-Time)
- สามารถทำการนับซ้ำต่อเนื่องด้วยการรับสัญญาณกระตุ้นจากภายนอก (Repetition Counter)
- มีอินพุตรับสัญญาณเบรก (Break) เพื่อหยุดการทำงานของตัวนับ
- สามารถซิงโครไนซ์กับสัญญาณนาฬิกาหรือสัญญาณกระตุ้นจากภายนอก รวมถึงสามารถซิงโครไนซ์ตัวไทมเมอร์ด้วยกันเองและไทมเมอร์ TIMx ได้
- ทำงานเชื่อมโยงกับ โมดูล DMA ทำให้สามารถนำค่าการนับหรือผลการทำงานของไทมเมอร์ส่งต่อไปยังโมดูลอื่นๆ ที่เชื่อมโยงผ่านโมดูล DMA ได้ โดยไม่ต้องส่งกลับไปประมวลผลที่ซีพียู
- สามารถสร้างสัญญาณอินเตอร์รัปต์จากการทำงานในแต่ละโหมดของไทมเมอร์ ประกอบด้วย
- การอินเตอร์รัปต์จากการปรับปรุงค่าเนื่องจากการนับค่าเกินหรือ โอเวอร์โฟลว์, การนับค่าต่ำเกินหรืออันเดอร์โฟลว์, การเตรียมความพร้อมของตัวนับด้วยซอฟต์แวร์หรือจากการกระตุ้นจากสัญญาณภายนอกและภายในชิพ
- การอินเตอร์รัปต์จากการรับสัญญาณกระตุ้นเพื่อเริ่มหรือหยุดการนับค่า
- การอินเตอร์รัปต์จากการทำงานในโหมดตรวจจับสัญญาณอินพุต
- การอินเตอร์รัปต์จากการเปรียบเทียบค่าในโหมดเอาต์พุตเปรียบเทียบข้อมูล
- การอินเตอร์รัปต์จากสัญญาณเบรก

2.1.9.2 การเลือกสัญญาณนาฬิกาของไทมเมอร์ TIMx

ไทมเมอร์ TIMx สามารถเลือกแหล่งกำเนิดสัญญาณนาฬิกาได้จาก 4 แหล่งเหมือนกับ TIM1 คือ

1. สัญญาณนาฬิกาภายใน (CK_INT)
2. สัญญาณนาฬิกาภายนอกโหมด 1: ใช้ขาอินพุตสัญญาณนาฬิกา รับสัญญาณนาฬิกาจากภายนอก
3. สัญญาณนาฬิกาภายนอกโหมด 2: ใช้ขาอินพุตทริกเกอร์ EUR รับสัญญาณนาฬิกาจากภายนอก
4. อินพุตทริกเกอร์ภายใน (ITRx): เป็นการที่ใช้ไทมเมอร์ภายในตัวหนึ่งทำหน้าที่เป็นปริสเกลเกอร์สำหรับไทมเมอร์อีกตัวหนึ่ง (ไม่ได้กล่าวถึงในบทนี้)

การทำงานและขั้นตอนในการเลือกแหล่งกำเนิดสัญญาณนาฬิกาของ TIMx เหมือนกับ TIM1 เพียงแต่เปลี่ยนรีจิสเตอร์ที่เกี่ยวข้องจาก TIM1 เป็น TIMx

2.1.9.3 รีจิสเตอร์ที่สำคัญของ TIMx ใน LPC1768

รีจิสเตอร์ที่สำคัญของ TIMx ใน LPC1768 มีดังนี้

1. TIMx_CR1 (Control Register 1): รีจิสเตอร์ควบคุมไทมเมอร์ TIMx ตัวที่ 1
2. TIMx_CR2 (Control Register 2): รีจิสเตอร์ควบคุมไทมเมอร์ TIMx ตัวที่ 2
3. TIMx_SMCR (Slave Mode Control Register): รีจิสเตอร์ควบคุมการทำงานโหมดสเลฟของไทมเมอร์ TIMx และใช้เลือกแหล่งกำเนิดสัญญาณนาฬิกา
4. TIMx_DIER (DMA and Interrupt Enable Register): รีจิสเตอร์เอ็นเอเบิล DMA และอินเตอร์รัปต์ของ TIMx
5. TIMx_SR (Status Register): รีจิสเตอร์สถานะของ TIMx
6. TIMx_EGR (Event Generation Register): รีจิสเตอร์แจ้งการเกิดเหตุการณ์ของ TIMx
7. TIMx_CCMR1 (Capture/Compare Mode Register 1): รีจิสเตอร์ควบคุมโหมดตรวจจับสัญญาณ/เปรียบเทียบข้อมูลของ TIMx ตัวที่ 1
8. TIMx_CCMR2 (Capture/Compare Mode Register 2): รีจิสเตอร์ควบคุมโหมดตรวจจับสัญญาณ/เปรียบเทียบข้อมูลของ TIMx ตัวที่ 2
9. TIMx_CCER (Capture/Compare Enable Register): รีจิสเตอร์เอ็นเอเบิลโหมดตรวจจับสัญญาณ/เปรียบเทียบข้อมูลของ TIMx
10. TIMx_CNT (Counter Register): รีจิสเตอร์ตัวนับของ TIMx
11. TIMx_PSC (Prescaler Register): รีจิสเตอร์กำหนดค่าปริสเกลเลอร์ของ TIMx
12. TIMx_ARR (Auto-Reload Register): รีจิสเตอร์ตั้งค่าการนับอัตโนมัติ
13. TIMx_CCR1 (Capture/Compare Register 1): รีจิสเตอร์เก็บค่าในโหมดตรวจจับสัญญาณ/เปรียบเทียบข้อมูลของ TIMx ตัวที่ 1
14. TIMx_CCR2 (Capture/Compare Register 2): รีจิสเตอร์เก็บค่าในโหมดตรวจจับสัญญาณ/เปรียบเทียบข้อมูลของ TIMx ตัวที่ 2
15. TIMx_CCR3 (Capture/Compare Register 3): รีจิสเตอร์เก็บค่าในโหมดตรวจจับสัญญาณ/เปรียบเทียบข้อมูลของ TIMx ตัวที่ 3
16. TIMx_CCR4 (Capture/Compare Register 4): รีจิสเตอร์เก็บค่าในโหมดตรวจจับสัญญาณ/เปรียบเทียบข้อมูลของ TIMx ตัวที่ 4
17. TIMx_DCR (DMA Control Register): รีจิสเตอร์ควบคุมการ DMA ของ TIMx

18. TIMx_DMAR (DAA Address for Burst Mode Register): รีจิสเตอร์กำหนดแอดเดรส DMA ของ TIMx

การทำงานและการแสดงผลการทำงานของรีจิสเตอร์ทั้งหมดของ TIMx เหมือนกับ TIM1 เพียงแต่เปลี่ยนชื่อจาก TIM1 เป็น TIMx โดย TIMx เป็นค่าที่เรียกรวมสำหรับไทมเมอร์ TIM2, TIM3 และ TIM4 ของ LPC1768F103 เมื่อ TIMx มี 3 ตัว ทำให้รีจิสเตอร์ที่เกี่ยวข้องทั้งหมดของ TIMx จึงมีแยกกันอีก 3 ชุดที่ทำงานเป็นอิสระต่อกัน อาจจะกล่าวได้ว่า LPC1768F103 จะมีไทมเมอร์ให้ใช้งานพร้อมกันได้มากถึง 4 ตัว (รวมนับ TIM1 ด้วย)

2.1.9.4 การใช้งานไทมเมอร์ TIMx ในการกำเนิดสัญญาณ PWM

สัญญาณ PWM (Pulse Width Modulation) ที่เกิดจากการ โมดูลไทมเมอร์ TIMx สามารถกำหนด ความถี่และความกว้างของสัญญาณหรือคิวตี้ไซเคิลได้ โดยมีหลักการทำงานดังนี้

1. ค่าความถี่ของสัญญาณ PWM กำหนดด้วย ค่าของรีจิสเตอร์รีโวลด์ค่าการนับอัตโนมัติหรือ TIMx_ARP
2. ค่าคิวตี้ไซเคิลของสัญญาณ PWM กำหนดด้วย ค่าของรีจิสเตอร์เก็บค่าในโหมดตรวจจับสัญญาณ/เปรียบเทียบข้อมูลของ TIMx ใดๆ หรือ TIMx_CCRx

การกำหนดให้ไทมเมอร์ TIMx ทำงานในโหมดกำเนิดสัญญาณ PWM สามารถกำหนดให้ทำงานพร้อมกันได้ เนื่องจากตัวนับแยกจากกันอย่างอิสระ ทำให้ LPC1768 สามารถกำเนิดสัญญาณ PWM ได้พร้อมกัน 3 ช่อง เมื่อใช้งานเฉพาะ TIMx และสูงถึง 8 ช่อง เมื่อใช้งาน TIM1 ร่วมด้วย สำหรับในหัวข้อนี้จะกล่าวถึงเฉพาะการกำเนิดสัญญาณ PWM ของไทมเมอร์ TIMx เท่านั้น

การเลือกโหมด PWM สำหรับ TIMx ทำได้โดยเขียนข้อมูล “101” หรือ “111” ไปยังบิต OCxM ในรีจิสเตอร์ TIMz_CCMRx ซึ่งมีด้วยกัน 2 โหมดคือ

i) PWM โหมด 1 - กำหนดค่าของบิต OCxM2 ถึง OCxM0 เป็น “110”: ในภาวะนั้นขึ้น จะเกิดสัญญาณ PWM ขึ้น ตรวจจับที่ค่าของรีจิสเตอร์ตัวนับ TIMx_CNT น้อยกว่าค่าของ TIMx_CCRx เมื่อนับค่าลง จะไม่มีการกำเนิดสัญญาณ PWM หากค่าของรีจิสเตอร์ตัวนับ TIMx_CNT ยังคงมากกว่าค่าของ TIMx_CCRx

ii) PWM โหมด 2 - กำหนดค่าของบิต OCxM2 ถึง OCxM0 เป็น “111”: การทำงานของโหมดนี้ ในภาวะนั้นขึ้นจะไม่มีการกำเนิดสัญญาณ PWM ตรวจจับที่ค่าของรีจิสเตอร์ตัวนับ TIMx_CNT น้อยกว่าค่าของ TIMx_CCRx ในภาวะนั้นค่าลง จะกำเนิดสัญญาณ PWM ตรวจจับที่ค่าของรีจิสเตอร์ตัวนับ TIMx_CNT ยังคงมากกว่าค่าของ TIMx_CCRx โดยโหมด 2 นี้จะการทำงานตรงข้ามกับโหมด 1

iii) เมื่อกำหนดโหมดของ PWM แล้วให้เอ็นเอเบิลการตั้งค่าลงในบิต OCxPE ในรีจิสเตอร์ TIMx_CCMRx ที่ตรงกับไทมเมอร์ TIMx ที่ต้องการให้กำเนิดสัญญาณ PWM และเซตบิต ARPE ในรีจิสเตอร์ TIMx_CR1 เพื่อทำการรีโหลดค่าการนับแบบอัตโนมัติ เมื่อทำงานในโหมดนับขึ้นและนับสู่ค่ากลาง

iv) การถ่ายทอค่าตั้งต้นการนับไปยังรีจิสเตอร์เงาจะเกิดขึ้นเมื่อมีเหตุการณ์ปรับปรุงค่าการนับขึ้นเท่านั้นดังนั้นจึงต้องทำการเซตบิต UG ในรีจิสเตอร์ TIMx_EGR จากนั้นทำการนับจะเริ่มคืบขึ้น

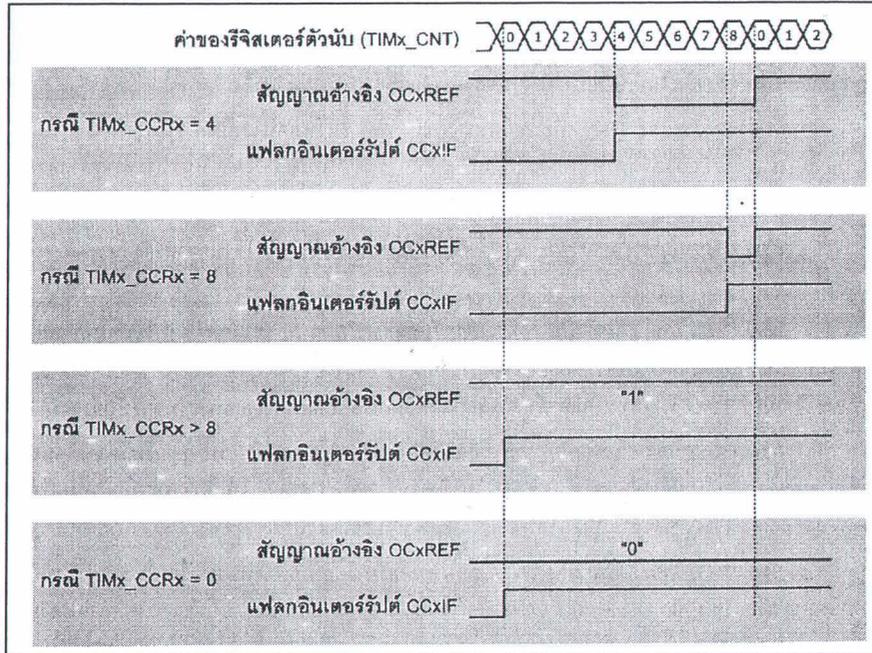
v) ขั้วของสัญญาณ PWM สามารถกำหนดให้เป็นสัญญาณ PWM แบบแอกทิฟลอจิกสูงหรือลอจิกต่ำได้ด้วยการเขียนข้อมูลไปยังบิต CCxP ในรีจิสเตอร์ TIMx_CCER จากนั้นทำการเอ็นเอเบิลของพอร์ตเอาต์พุตที่จะใช้ส่งสัญญาณ PWM ออกไปด้วยการกำหนดค่าไปยังบิต CCxE ในรีจิสเตอร์ TIMx_CCER

vi) ในขณะที่ทำงาน ค่าของรีจิสเตอร์ตัวนับ TIMx_CNT และค่าของรีจิสเตอร์เก็บค่า TIMx_CCRx จะถูกเปรียบเทียบและตรวจสอบตลอดเวลาว่า ค่าของ TIMx_CCRx น้อยกว่าหรือเท่ากับค่าของรีจิสเตอร์ตัวนับ TIMx_CNT หรือค่าของ TIMx_CNT น้อยกว่าค่าของรีจิสเตอร์เก็บค่า TIMx_CCRx ทั้งนี้ขึ้นกับการกำหนดทิศทางในการนับ

vii) การนับค่าของไทมเมอร์ TIMx เพื่อกำเนิดสัญญาณ PWM มี 2 โหมด คือ โหมดปรับเทียบขอบสัญญาณแบบนับค่าขึ้น และโหมดปรับเทียบกึ่งกลางสัญญาณ สามารถเลือกได้ด้วยการกำหนดค่าไปยังบิต CMS1 และ CMS0 ในรีจิสเตอร์ TIMx_CR1

2.1.9.5 การเกิดสัญญาณ PWM ในโหมดปรับเทียบขอบสัญญาณแบบนับค่าขึ้น

การนับค่าขึ้นของไทมเมอร์ TIMx เกิดขึ้นเมื่อบิต DIR ในรีจิสเตอร์ TIMa_CR1 เป็น “0” หากเลือกให้กำเนิดสัญญาณ PWM โหมด 1 สัญญาณ PWM อ้างอิง หรือ OCxREF (ใช้อ้างอิงเท่านั้น ไม่มีสัญญาณ) OCxREF เกิดขึ้นในทางปฏิบัติ การคงอยู่ของสัญญาณ OCxREF หมายถึง ยังมีการกำเนิดสัญญาณ PWM อยู่ที่ขาของพอร์ตของไมโครคอนโทรลเลอร์) จะเป็นลอจิกสูงตราบเท่าที่ค่าของรีจิสเตอร์ตัวนับ TIMx_CNT น้อยกว่าค่าของรีจิสเตอร์ TIMx_CCRx ถ้าหากค่าของ TIMx_CNT เท่ากับหรือมากกว่าค่าของรีจิสเตอร์ TIMx_CCRx สัญญาณอ้างอิง OCxREF จะเปลี่ยนเป็นลอจิกต่ำ ดังแสดงในรูปที่ 2.9 จากในภาพ กำหนดค่าของรีจิสเตอร์ TIMx_ARR เท่ากับ 8 ถ้าหากกำหนดค่าของ TIMx_CCRx สูงกว่าค่าของ TIMx_ARR สัญญาณอ้างอิง OCxREF จะเป็นลอจิกสูงหรือ “1” ตลอดเวลา และในทางกลับกันหากกำหนดค่าของ TIMx_CCRx เป็น 0 สัญญาณอ้างอิง OCxREF จะเป็นลอจิกต่ำหรือ “0” ตลอดเวลา เช่นกัน ดังนั้นหากต้องการให้มีการกำเนิดสัญญาณ PWM อย่างถูกต้องควรกำหนดค่าของ TIMx_CCRx ให้มีค่าอยู่ระหว่าง 1 ถึง TIMx_ARR



รูปที่ 2.9 การทำงานเพื่อสร้างสัญญาณ PWM ในโหมดปรับเทียบขอบขาสัญญาณแบบนับค่าขึ้น

2.1.9.6 การกำเนิดสัญญาณ PWM ในโหมดปรับเทียบกึ่งกลางสัญญาณ

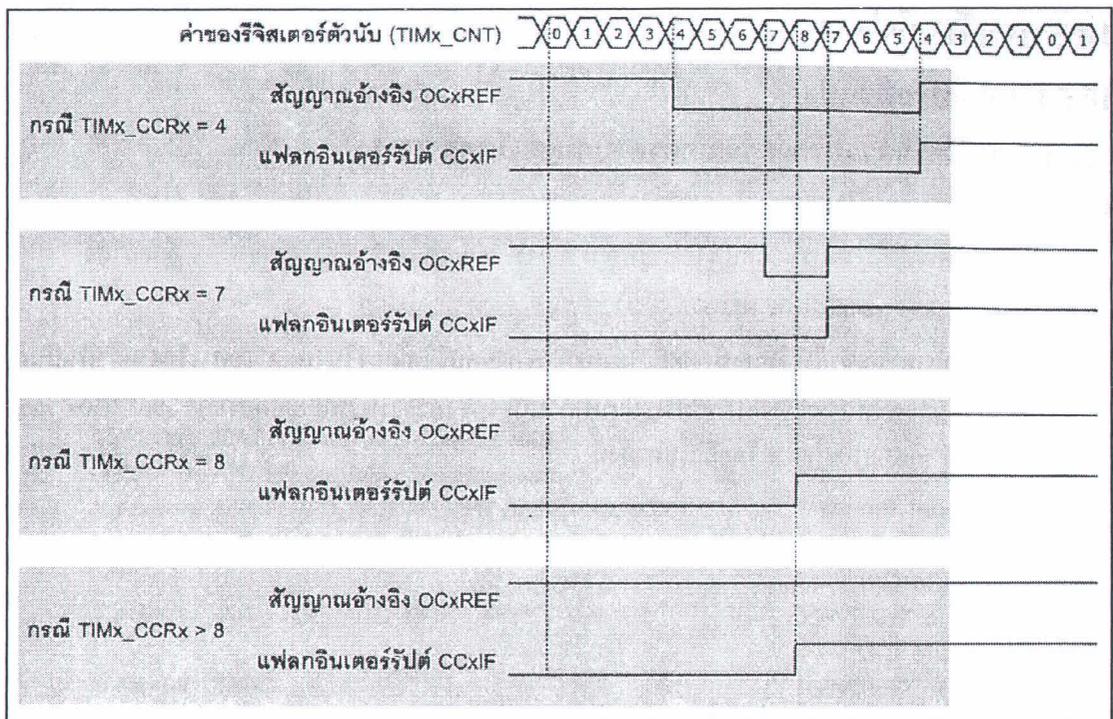
การทำงานในโหมดนี้จะเกิดขึ้นเมื่อบิต CMS1 และ CMS0 ในรีจิสเตอร์ TIMx_CR1 เป็นค่าใดๆที่ไม่ใช่ “00” บิตแฟลกเปรียบเทียบจะถูกกำหนดให้เซตเมื่อตัวนับเริ่มนับขึ้น, นับค่าลง หรือทั้งสองกรณี ขึ้นอยู่กับโหมดการทำงานที่กำหนดไปยังบิต CMS1 และ CMS0 ในรีจิสเตอร์ TIMx_CR1 บิตกำหนดทิศทาง การนับจะถูกเปลี่ยนแปลงอย่างอัตโนมัติตามการทำงานของฮาร์ดแวร์

การนับค่าของไทมเมอร์ TIMx เพื่อกำเนิดสัญญาณ PWM ในโหมดนี้จะเป็นการนับสู่ค่ากลาง นั่นคือ จะเริ่มด้วยการนับขึ้นจนถึงค่าที่กำหนดก็จะเปลี่ยนทิศทางการนับเป็นนับลง โดยโหมดการนับสู่ค่ากลางยังสามารถกำหนดได้อีก 3 แบบจากการกำหนดค่าบิต CMS1 และ CMS0 ในรีจิสเตอร์ TIMx_CR1 ดังนี้

1. กรณี CMS1 และ CMS0 เป็น “00” – เลือกโหมดการนับขึ้น (ไม่ใช้ในการสร้าง PWM)
2. กรณี CMS1 และ CMS0 เป็น “01” – เลือกโหมดการนับสู่ค่ากลาง แบบที่ 1: ในแบบนี้ ตัวนับจะนับค่าขึ้นและลงสลับกัน บิต CCxS1 และ CCxS0 ในรีจิสเตอร์ TIMx_CCMRx ให้เป็น “00” เพื่อกำหนดให้ขาพอร์ตของไทมเมอร์ TIMx ที่ใช้งานเป็นเอาต์พุต และบิตแฟลกเปรียบเทียบจะเซตเฉพาะในช่วงนับค่าลง

3. กรณี CMS1 และ CMS0 เป็น “10” – เลือกโหมดการนับสู่ค่ากลาง แบบที่ 2: ในแบบนี้ ตัวนับจะนับค่าขึ้นและลงสลับกัน บิต CCxSI และ CCxSO ในรีจิสเตอร์ TIMx_CCMRx ให้เป็น “00” เพื่อกำหนดให้ขาพอร์ตของไทมเมอร์ TIMx ที่ใช้งานเป็นเอาต์พุต และบิตเฟล็กเปรียบเทียบ จะเซตเฉพาะในช่วงนับค่าขึ้น
4. กรณี CMS1 และ CMS0 เป็น “11” – เลือกโหมดการนับสู่ค่ากลาง แบบที่ 3: ในแบบนี้ ตัวนับจะนับค่าขึ้นและลงสลับกัน บิต CCxSI และ CCxSO ในรีจิสเตอร์ TIMx_CCMRx ให้เป็น “00” เพื่อกำหนดให้ขาพอร์ตของไทมเมอร์ TIMx ที่ใช้งานเป็นเอาต์พุต และบิตเฟล็กเปรียบเทียบจะเซตทั้งช่วงนับค่าลงและขึ้น

ในรูปที่ 2.10 แสดงการทำงานเพื่อสร้างสัญญาณ PWM ในโหมดปรับเทียบกึ่งกลางสัญญาณ โดยกำหนดค่าของรีจิสเตอร์ TIMx_ARR เท่ากับ 8 และเลือกกำเนิดสัญญาณ PWM ในโหมด 1 นั่นคือ จะมีสัญญาณ PWM เกิดขึ้นเมื่อค่าของรีจิสเตอร์ตัวนับ TIMx_CNT น้อยกว่าค่าของ TIMx_CCRx และเลือกการนับสู่ค่ากลางแบบที่ 1 ดังนั้นบิตเฟล็กจะเซตเมื่อตัวนับทำการนับค่าลงจนถึงเท่ากับหรือน้อยกว่าค่าการนับที่ตั้งไว้ในรีจิสเตอร์ TIMx_ARR



รูปที่ 2.10 การทำงานเพื่อสร้างสัญญาณ PWM ในโหมดปรับเทียบกึ่งกลางสัญญาณ



2.2 ทฤษฎีวงจรไฟฟ้าพื้นฐาน^{1,2}

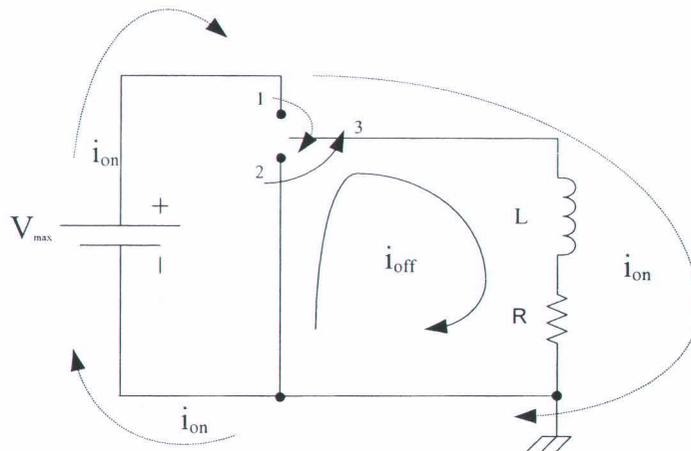
ในหัวข้อนี้จะนำเสนอวงจรไฟฟ้าแบบใหม่ที่น่าสนใจในงานวิจัยฉบับนี้เพื่อทำหน้าที่เป็นวงจรเปิด/ปิดพัลส์ โดยจะกล่าวถึงทฤษฎีวงจรไฟฟ้าพื้นฐาน¹ ที่ใช้วิเคราะห์กระแสไฟฟ้าที่เกิดขึ้นในวงจรเปิด/ปิดพัลส์ที่ได้ทำการออกแบบ โดยในรูปที่ 2.11 แสดงวงจรไฟฟ้าที่ทำหน้าที่เป็นวงจรเปิด/ปิดพัลส์แบบใหม่ที่น่าสนใจ และตัวเลข 1, 2 และ 3 เป็นตัวเลขกำกับขาของทรานซิสเตอร์ โดยในที่นี้ เรานำทรานซิสเตอร์มาทำงานในโหมด Switching และต่อใช้งานเป็นแบบ Common Emitter เพื่อทำหน้าที่เปิด/ปิดกระแสไฟฟ้าที่ไหลในวงจรเสมือนว่าเป็นการชาร์ตและดีสชาร์ตกระแสไฟฟ้าที่ไหลเข้า/ออกจากขดลวด ตามลำดับ ดังนั้นเราสามารถวิเคราะห์การทำงานของวงจрдังกล่าวได้ 2 กรณี คือ กรณีที่ 1 Contact 1 กับ 3 ต่อกัน (ปิดวงจร) กระแสไฟฟ้าที่ไหลมีค่า i_{on} และ กรณีที่ 2 Contact 2 กับ 3 ต่อกัน (เปิดวงจร) กระแสไฟฟ้าที่ไหลมีค่า i_{off}

2.2.1 วิเคราะห์การทำงานในสภาวะปิดวงจร

รูปที่ 2.12 ทางซ้ายแสดงวงจรไฟฟ้ารูปที่ 2.11 ในสภาวะที่ Contact 1 กับ 3 เชื่อมต่อกัน (ปิดวงจร) โดยจะหากระแสไฟฟ้าที่ไหลในวงจร i_{on} ได้จากการพิจารณาใช้กฎ KVL ดังนี้

$$V_{max} - L \frac{dI}{dt} = IR \quad (2.1)$$

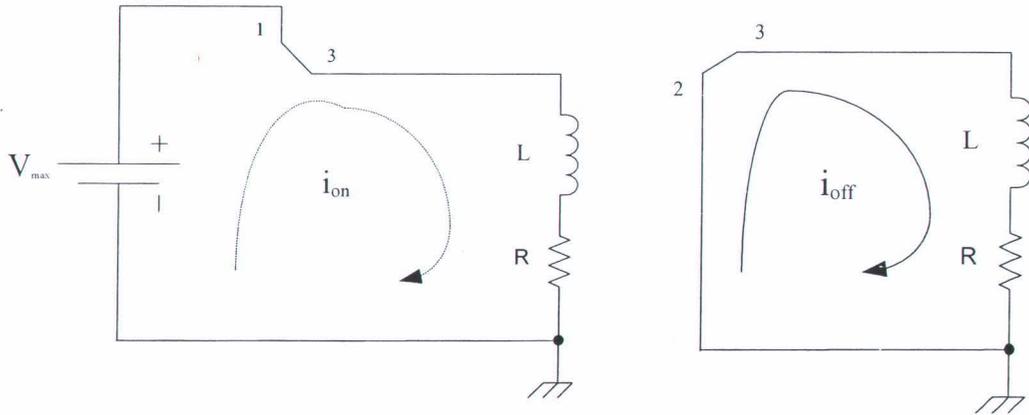
$$I = \frac{V_{max}}{R} (1 - e^{-\frac{Rt}{L}}) \quad (2.2)$$



รูปที่ 2.11 วงจรไฟฟ้าที่ทำหน้าที่เป็นวงจรเปิด/ปิดพัลส์แบบใหม่ที่น่าสนใจ

¹ S.R. Paranjothi. (2010). Electric Circuit Analysis. (3rd ed.). New Age Science.

² สุทัศน์ แสงกระจ่าง. (2553). วงจรเปิด/ปิดพัลส์และการโปรแกรมสำหรับตรวจจับผลตอบสนองเชิงเวลาของกระแสไฟฟ้าในขดลวดที่เกิดจากโลหะ โดยใช้การเหนี่ยวนำด้วยพัลส์. มหาวิทยาลัยธุรกิจบัณฑิตย์. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต.



รูปที่ 2.12 สถานะที่ Contact 1 กับ 3 ต่อกัน (ปิดวงจร) และ สถานะที่ Contact 2 กับ 3 ต่อกัน (เปิดวงจร)

$$(2.3) \quad \text{ดังนั้น} \quad i_{on} = I_0(1 - e^{-\frac{t}{\tau}}) \quad \text{โดยที่} \quad \tau = \frac{L}{R} \quad \text{และ} \quad I_0 = \frac{V_{max}}{R}$$

2.2.2 วิเคราะห์การทำงานในสถานะเปิดวงจร

รูปที่ 2.12 ทางขวาแสดงวงจรไฟฟ้ารูปที่ 2.11 ในสถานะที่ Contact 2 กับ 3 เชื่อมต่อกัน (เปิดวงจร) โดยจะหากระแสไฟฟ้าที่ไหลในวงจร i_{off} ได้จากการพิจารณาใช้กฎ KVL ดังนี้

$$L \frac{di}{dt} = IR \quad (2.4)$$

$$I_{off} = e^{-\frac{Rt}{L}} \quad (2.5)$$

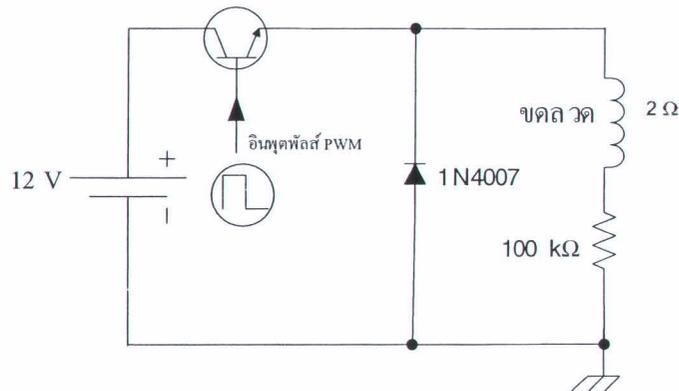
$$(2.6) \quad \text{ดังนั้น} \quad i_{off} = e^{-\frac{t}{\tau}} \quad \text{โดยที่} \quad \tau = \frac{L}{R}$$

จากการวิเคราะห์ห้วงจรไฟฟ้าทั้ง 2 กรณี พบว่า ปัจจัยที่มีผลคือค่าความต้านทาน (R) และค่าความเหนี่ยวนำ (L) จากการทดลองเพื่อวัดค่าเหนี่ยวนำที่เกิดขึ้นจากวงจรไฟฟ้างดกล่าว โดยนำโลหะมาวางอยู่ห่างจากขดลวด ณ ระยะห่างค่าต่างๆ กัน แล้วทำการวัดค่า L เครื่องมือวัดค่า RLC พบว่า ค่า L ที่วัดได้เกิดการเปลี่ยนแปลง สาเหตุหลักที่ทำให้ค่า L เกิดการเปลี่ยนแปลงเมื่อมีวัตถุโลหะมาอยู่ใกล้กับขดลวดนั้น สามารถที่จะอธิบายได้โดยอาศัยทฤษฎีสวนแม่เหล็กดังนี้ “เมื่อขดลวดอยู่ใกล้กับโลหะ ค่า μ_r (ค่า permeability ของขดลวด) มีค่าลดลง ในทางกลับกัน ถ้าวัตถุอยู่ห่างออกไปจากโลหะไปเรื่อยๆ ค่า μ_r จะมีค่าที่เพิ่มขึ้น จนกระทั่งค่า μ_r มีค่าเท่ากับ μ_{air} (ค่า permeability ของอากาศ)” ซึ่งถ้าเลือกขดลวด (ตัวเหนี่ยวนำหรือสัญลักษณ์ L ในวงจรรูปที่ 2.11) เบอร์ใหญ่ขึ้น แต่จำนวนรอบในการพันมีค่าเท่าเดิม จะทำให้

ค่าความเหนี่ยวนำมีค่าที่สูงขึ้น ความต้านทานในวงจรมีค่าลดลง ค่า $\Delta T_{discharge}$ มีค่ามากขึ้น จึงส่งผลดีต่อวงจรเปิด/ปิดพัลส์แบบใหม่ที่น่าเสนอคือ จะทำให้วงจรมีความไวในการตรวจจับโลหะได้ดียิ่งขึ้น

2.3 การใช้งานทรานซิสเตอร์ร่วมกับพัลส์ PWM

ในหัวข้อนี้จะกล่าวถึงการใช้งานทรานซิสเตอร์ร่วมกับพัลส์ PWM (Pulse Width Modulation) ในงานวิจัยฉบับนี้ จึงทำให้วงจรรูปที่ 2.11 ถูกแก้ไขใหม่โดยวางทรานซิสเตอร์และไดโอดเข้าไปเพิ่มเติมในวงจรดังแสดงในรูปที่ 2.13 โดยทรานซิสเตอร์ถูกกำหนดให้ทำงานภายใต้สภาวะที่กำหนด 2 ลักษณะ คือ เปิดวงจรกับปิดวงจร ซึ่งเป็นไปตามหลักการการทำงานโดยปกติของทรานซิสเตอร์อยู่แล้ว ซึ่งการทำงานใน 2 ลักษณะดังกล่าวนี้ขึ้นอยู่กับระดับแรงดันของพัลส์ PWM ที่ป้อนให้แก่ทรานซิสเตอร์



รูปที่ 2.13 วงจรไฟฟ้าที่พัฒนาเพิ่มขึ้นจากวงจรรูปที่ 2.11 โดยมีการต่อทรานซิสเตอร์และไดโอด

โดยหลักการการทำงานและผลกระทบที่ได้รับจากสภาวะที่ทรานซิสเตอร์ทำงาน (สภาวะปิดวงจร) และไม่ทำงาน (สภาวะเปิดวงจร) จะไม่ขอกล่าวถึงในที่นี้ ผู้สนใจสามารถอ่านเพิ่มเติมได้จากเอกสารอ้างอิง¹ แต่สรุปได้ว่า เมื่อกระแสไฟฟ้าผ่านขดลวด (Coil) จะเกิดสนามแม่เหล็กไปเหนี่ยวนำโลหะ ทำให้เกิด Eddy Current บนผิวของโลหะ ซึ่งส่งผลต่อค่าความเหนี่ยวนำในเชิงวงจรไฟฟ้าด้วยเช่นกัน โดยจะเป็นการส่งต่อพลังงานจากขดลวดไปสู่ผิวของโลหะที่อยู่ใกล้ๆ ซึ่งชนิดโลหะต่างกันจะมีความต้านทานที่ต่างกันและส่งผลให้ค่า $T_{discharge}$ ต่างกันด้วย โดย $T_{discharge}$ เป็นค่าผลต่างของค่าเวลาดั้งแต่จุดเวลาที่กระแสไฟฟ้าในขดลวดเริ่มที่จะลดลง (เริ่มการส่ง Pulse Off) ไปจนถึงเวลาที่ค่าแรงดันตกคร่อมไดโอดมีค่าเท่ากับ 0

¹ สุทัศน์ แยกระจ่าง. (2553). วงจรเปิด/ปิดพัลส์และการโปรแกรมสำหรับตรวจจับผลตอบสนองเชิงเวลาของกระแสไฟฟ้าในขดลวดที่เกิดจากโลหะ โดยใช้การเหนี่ยวนำด้วยพัลส์. มหาวิทยาลัยธุรกิจบัณฑิตย์. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต.

ตารางที่ 2.1 คุณสมบัติทางกายภาพในทางทฤษฎีที่คำนวณได้ของขดลวดที่มีรูปทรงและขนาดต่างๆ²

Size	Shape	Turns	Wire size	Inductance	Resistance
Ø 120 mm	Round	36	Ø 0.40 mm / 0.14 mm ²	405 µH	1.9 Ohm
Ø 150 mm	Round	31	Ø 0.40 mm / 0.14 mm ²	394 µH	2.0 Ohm
Ø 175 mm	Round	28	Ø 0.40 mm / 0.14 mm ²	387 µH	2.1 Ohm
Ø 200 mm	Round	26	Ø 0.40 mm / 0.14 mm ²	406 µH	2.2 Ohm
Ø 250 mm	Round	22	Ø 0.40 mm / 0.14 mm ²	380 µH	2.3 Ohm
Ø 300 mm	Round	20	Ø 0.50 mm / 0.20 mm ²	390 µH	1.6 Ohm
Ø 400 mm	Round	17	Ø 0.50 mm / 0.20 mm ²	396 µH	1.8 Ohm
Ø 500 mm	Round	15	Ø 0.50 mm / 0.20 mm ²	400 µH	2.0 Ohm
1.0 x 1.0 m	Square	10	Ø 0.66 mm / 0.34 mm ²	406 µH	2.0 Ohm
1.4 x 1.4 m	Square	8	Ø 0.66 mm / 0.34 mm ²	387 µH	2.2 Ohm
1.8 x 1.8 m	Square	7	Ø 0.80 mm / 0.50 mm ²	398 µH	1.7 Ohm

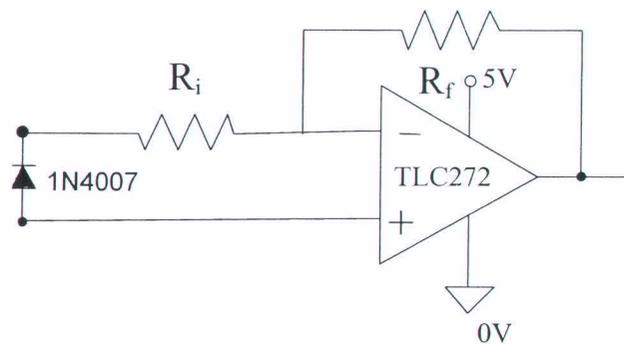
2.4 ทฤษฎีการพันเส้นลวดเป็นรูปวงกลม

จากการศึกษาค้นคว้า พบว่า ในงานวิจัย¹ ได้ทำการศึกษาเกี่ยวกับเครื่องตรวจจับโลหะที่อาศัยหลักการเหนี่ยวนำด้วยพัลส์โดยใช้การประมวลผลสัญญาณแบบดิจิทัล จากงานวิจัยดังกล่าวได้นำเส้นลวดที่มีขนาดเส้นผ่านศูนย์กลางตั้งแต่ 0.40 – 0.80 mm โดยมีค่าคุณสมบัติทางกายภาพในทางทฤษฎีที่คำนวณได้ดังแสดงในตารางที่ 2.1 ซึ่งถึงแม้ว่าขนาดและรูปร่างทางกายภาพของขดลวดจะนำมาพันเป็นรูปวงกลมหรือสี่เหลี่ยมที่แตกต่างกันในการใช้งานก็ตาม แต่ค่าความเหนี่ยวนำและความต้านทานของขดลวดแต่ละแบบนั้นก็ยังมีค่าแตกต่างกันเพียงเล็กน้อยเท่านั้น ซึ่งมีค่าอยู่ระหว่าง 380 – 406 µH และ 1.6 – 2.3 โอห์ม ตามลำดับ โดยขดลวดที่มีขนาดเล็กกว่า ย่อมจะต้องมีจำนวนรอบของขดลวดที่มากขึ้นด้วย ซึ่งโดยทั่วไป ค่าความเหนี่ยวนำที่เหมาะสมและเป็นที่ยอมรับใช้งานในเครื่องตรวจจับโลหะที่ใช้หลักการการเหนี่ยวนำด้วยพัลส์ (Pulse Induction Metal Detector) จะมีค่าอยู่ระหว่าง 300 – 500 µH ซึ่งในงานวิจัย² นี้ก็ได้เลือกใช้ค่าความเหนี่ยวนำของขดลวดที่ใช้ในการทดลองเพื่อตรวจจับโลหะที่มีค่าเท่ากับ 400 µH ดังนั้น ในงานวิจัยที่นำเสนอฉบับนี้จึงได้นำเส้นลวดที่มีขายตามท้องตลาดที่มีเส้นผ่านศูนย์กลางขนาด 0.40 mm หรือมีพื้นที่หน้าตัดประมาณ 0.14 mm² และมีจำนวนหุ้มมาพันเป็นขดลวดรูปวงกลมขนาดเส้นผ่านศูนย์กลางเท่ากับ 15 cm และกำหนดจำนวนรอบที่พันเป็นเลขลงตัวเท่ากับ 35 รอบ และมีความหนาของขดลวดประมาณ 1 cm เพื่อให้มีคุณสมบัติในการเลือกใช้งานตรงกับกรณีที่ 2 ของตารางที่ 2.1

¹ Pulse induction metal detector with DSP. from http://www.lammertbies.nl/electronics/PI_metal_detector.html.

² Single-layer Coil Inductance and Q. from <http://www.qsl.net/in3otd/indcalc.html>.

หลังจากนั้น ทำการวัดค่าความเหนี่ยวนำและค่าความต้านทานของขดลวดที่ใช้ทดสอบดังกล่าว พบว่า มีค่าความเหนี่ยวนำเท่ากับ $420 \mu\text{H}$ และมีค่าความต้านทานเท่ากับ 2 โอห์ม ซึ่งเป็นค่าความเหนี่ยวนำและความต้านทานของขดลวดที่ใกล้เคียงกับค่าที่ปรากฏในกับกรณีที่ 2 ของตารางที่ 2.1 จริง อีกทั้งเป็นค่าความเหนี่ยวนำที่ใกล้เคียงกับงานวิจัยที่ผ่านมาด้วย ซึ่งเลือกใช้ค่าความเหนี่ยวนำของขดลวดเท่ากับ $400 \mu\text{H}$ ซึ่งในที่นี้ อาจะกำหนดจำนวนรอบของการพันเส้นลวดเป็นขดลวดที่เป็นรูปทรงอื่นๆ หรือมีจำนวนรอบค่าอื่นๆ ก็ได้ แต่สิ่งที่สำคัญที่จะต้องคำนึงถึงเสมอในการใช้งานขดลวดในการตรวจจับโลหะก็คือ ค่าความเหนี่ยวนำและความต้านทานของขดลวดที่ใช้งานจะต้องมีค่าที่ใกล้เคียงกับ $400 \mu\text{H}$ และ 2 โอห์ม ตามลำดับ



รูปที่ 2.14 วงจรไฟฟ้ารูปที่ 2.5 ในสถานะที่ Contact 1 กับ 3 เชื่อมต่อกัน (ปิดวงจร)

หัวข้อที่ 2.5 และ 2.6 ต่อไปนี้เป็นเนื้อหาที่นำมาจากงานวิจัย¹ ซึ่งเป็นการใช้งาน AVR ATmega168 เพื่อที่จะอธิบายหลักการพื้นฐานที่สำคัญที่ใช้ในการออกแบบวงจรเปิด/ปิดพัลส์ที่นำเสนอในงานวิจัยฉบับนี้

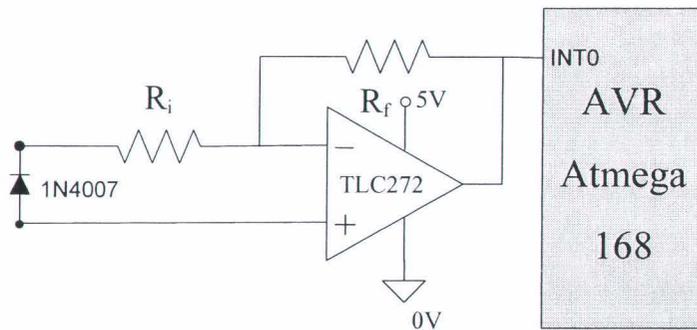
2.5 การตรวจจับค่าแรงดันตกคร่อมไดโอดมีค่าเท่ากับศูนย์

ในหัวข้อนี้จะกล่าวถึงหลักการทำงานของการตรวจจับค่าแรงดันตกคร่อมไดโอดมีค่าเท่ากับ 0 โดยนำออปแอมป์มาต่อใช้งานเพื่อขยายสัญญาณอินพุต ดังรูปที่ 2.14 โดยจะเลือกไดโอดที่ทนรับกระแสไฟฟ้าได้มากกว่าค่า Ampere Maximum ของวงจรเปิด/ปิดพัลส์ที่มีค่าเท่ากับ 6 แอมแปร์และในวงจรที่ออกแบบนี้จะมีค่าแรงดันตกคร่อมไดโอด (V_d) ประมาณ 0.3 – 0.8 โวลต์ แล้วแต่ชนิดของไดโอดที่ใช้งาน ในที่นี้เราเลือกใช้ใช้งานไดโอดเบอร์ 1N4007 ซึ่งเป็นไดโอดที่มีค่า t_{on} กับ t_{off} ต่ำ และมีค่า V_d ประมาณ 0.65 โวลต์ ทั้งนี้จะเลือกใช้ไดโอดเบอร์ใดก็ได้ แต่ไดโอดดังกล่าวจะต้องมีค่าทำงานโดยมีค่า t_{on} กับ t_{off} ที่ต่ำ เพื่อที่ว่าวงจรเปิด/ปิดพัลส์จะสามารถทำงานได้อย่างรวดเร็วตามต้องการ รวมถึงมีค่า V_d อยู่ระหว่าง 0.3 – 0.8 โวลต์

¹ สุทัศน์ แขกระจำง. (2553). วงจรเปิด/ปิดพัลส์และการโปรแกรมสำหรับตรวจจับผลตอบสนองเชิงเวลาของกระแสไฟฟ้าในขดลวดที่เกิดจาก

โลหะโดยใช้การเหนี่ยวนำด้วยพัลส์. มหาวิทยาลัยธุรกิจบัณฑิต. วิทยาลัยวิศวกรรมศาสตรมหาบัณฑิต.

เหตุผลที่สำคัญที่เลือกอัตราขยายของออปแอมป์มีค่าเท่ากับ 25 เท่า ก็เพราะว่า “เราต้องการสร้างสัญญาณเอาต์พุตของวงจรเปิด/ปิดพัลส์ (ขาเอาต์พุตของออปแอมป์) ที่มีการเปลี่ยนแปลงของระดับแรงดันไฟฟ้า จากระดับแรงดันไฟฟ้า 5 โวลต์ไปเป็นระดับแรงดันไฟฟ้า 0 โวลต์ เปรียบเสมือนเป็น “สัญญาณพัลส์ขอบขาลง” ที่ชัดเจน ซึ่งเราสามารถนำสัญญาณพัลส์ขอบขาลงดังกล่าวไปกระตุ้นขาอินเทอร์รัปต์ภายนอกช่องที่ 0 (INT0) ของ AVR ATmega168 เพื่อให้เกิดการอินเทอร์รัปต์และทำงานเพื่อตรวจจับระยะเวลาที่เกิดผลตอบสนองตัดศูนย์กลางของกระแสไฟฟ้าในขดลวดได้ ดังแสดงในรูปที่ 2.13



รูปที่ 2.15 วงจรที่ต่อเอาต์พุตของออปแอมป์เข้าขาอินเทอร์รัปต์ภายนอกช่องที่ 0 ของ ATmega168

จากการที่เราใช้ออปแอมป์ TLC272 มาต่อใช้งานแบบ Inverting Amplifier และกำหนดให้ระดับแรงดันอินพุต V_{in} และระดับแรงดันอินพุต V_{out} มีค่าเท่ากับ 0.2 โวลต์ และ 5 โวลต์ตามลำดับ นั้น เราสามารถที่จะหาค่าของ R_i กับ R_f เพื่อนำมาใช้งานวงจรเปิด/ปิดพัลส์ที่นำเสนอตั้งในรูปที่ 2.15 ได้โดยใช้สมการที่ 2.7 – 2.9 ดังนี้

$$V_{out} = V_{in} \left(1 + \frac{R_f}{R_i} \right) \quad (2.7)$$

$$5 = 0.2 * \left(1 + \frac{R_f}{R_i} \right) \quad (2.8)$$

$$\frac{R_f}{R_i} = 24 \quad (2.9)$$

ในที่นี้ ถ้าเราเลือกใช้งานตัวต้านทาน R_i มีค่าเท่ากับ 1 k Ω และนำค่า R_f นี้ไปแทนลงในสมการที่ (2.9) เราได้ตัวต้านทาน R_f มีค่าเท่ากับ 24 k Ω เช่นเดียวกับงานวิจัย¹

¹ สุทัศน์ แยกกระจ่าง. (2553). วงจรเปิด/ปิดพัลส์และการโปรแกรมสำหรับตรวจจับผลตอบสนองเชิงเวลาของกระแสไฟฟ้าในขดลวดที่เกิดจาก

โลหะโดยใช้การเหนี่ยวนำด้วยพัลส์. มหาวิทยาลัยธุรกิจบัณฑิต. วิทยาลัยวิศวกรรมศาสตรมหาบัณฑิต.

ดังนั้น จะเห็นได้ว่า หลักการทำงานของการทำงานของวงจรจับค่าแรงดันตกคร่อมไดโอดมีค่าเท่ากับ 0 โดยนำ ออปแอมป์มาต่อใช้งานเพื่อขยายสัญญาณอินพุตแบบ Inverting Amplifier ดังรูปที่ 2.15 นั้นสามารถนำมาใช้งานได้ในงานวิจัยฉบับนี้ เช่น โดยสัญญาณเอาต์พุตของออปแอมป์มีการเปลี่ยนแปลงระดับแรงดันไฟฟ้าอยู่ระหว่าง 5 โวลต์กับ 0 โวลต์อย่างชัดเจน ซึ่งเปรียบเสมือนเป็นสัญญาณพัลส์ที่สามารถนำไปกระตุ้นขาอินเทอร์รัปต์ภายนอกช่องที่ 0 ของ AVR ATmega168 เพื่อให้เกิดการอินเทอร์รัปต์และ AVR ATmega168 ตรวจสอบระยะเวลาที่เกิดผลตอบสนองตัดศูนย์กลางของกระแสไฟฟ้าในขดลวดได้อย่างมีประสิทธิภาพและเชื่อถือได้

2.6 วงจรเปิด/ปิดสัญญาณพัลส์ให้แก่ขดลวด

โดยทั่วไป วงจรเปิด/ปิดพัลส์ให้แก่ขดลวด (Coil Driver) ที่มีใช้งานในเครื่องตรวจจับโลหะนั้น^{1,2} มีหน้าที่หลักที่สำคัญคือทำหน้าที่เปิดและปิดสัญญาณพัลส์ PWM ให้แก่ขดลวดและส่งผ่านผลตอบสนองเชิงเวลาของขดลวดของกระแสไฟฟ้าในขดลวดที่ตรวจจับได้จากโลหะแต่ละชนิดที่ทดสอบไปยังวงจรแอนะล็อกที่ทำจากอุปกรณ์ RLC หลังจากนั้น จะนำค่าแรงดันไฟฟ้าหรือกระแสไฟฟ้าไปประมวลผลและระบุชนิดของโลหะที่ตรวจจับได้ต่อไป โดยเครื่องตรวจจับโลหะที่ขายโดยทั่วไปจะมีวงจรเปิด/ปิดสัญญาณพัลส์ให้แก่ขดลวดที่แตกต่างกัน ขึ้นอยู่กับราคาและคุณภาพของเครื่องตรวจจับโลหะ

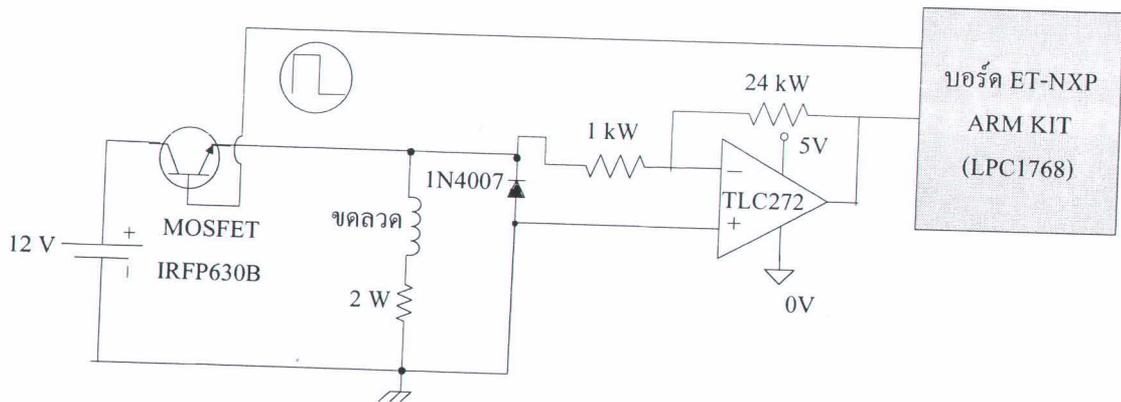
ในงานวิจัยฉบับนี้จะออกแบบและนำเสนอวงจรอิเล็กทรอนิกส์เพื่อทำหน้าที่เปิด/ปิดสัญญาณพัลส์ให้แก่ขดลวด (Coil Driver) ขึ้นมาใหม่โดยอ้างอิงหลักการเหนี่ยวนำด้วยพัลส์และวิธีการใช้งานในทางปฏิบัติ อีกทั้งยังนำไมโครคอนโทรลเลอร์มาใช้ในการควบคุมการทำงานและสร้างสัญญาณพัลส์ PWM อีกด้วย ซึ่งในที่นี้ จะนำรูปวงจรที่ 2.8 และ 2.11 มาเชื่อมต่อกันเพื่อที่จะทำหน้าที่เป็นวงจรเปิด/ปิดพัลส์ให้แก่ขดลวด (ตัว L) ประโยชน์ที่ได้รับจากการใช้วงจรเปิด/ปิดพัลส์ใหม่นี้ที่นำเสนอคือ การออกแบบวงจรกระทำได้ง่าย มีเสถียรภาพและประสิทธิภาพสูง บำรุงรักษาง่าย โดยลดความยุ่งยากในการปรับจูนวงจรแอนะล็อกแบบเดิมที่ใช้งานกันอยู่ในปัจจุบันนี้

วงจรเปิด/ปิดพัลส์ (Coil Driver) ทำหน้าที่เปิดและปิดสัญญาณพัลส์ PWM ให้แก่ขดลวดและส่งผ่านผลตอบสนองเชิงเวลาตัดศูนย์กลางของกระแสไฟฟ้าในขดลวดที่ตรวจจับได้จากโลหะแต่ละชนิดที่ทดสอบไปยังไมโครคอนโทรลเลอร์ โดยรูปวงจรที่นำมาเสนอเป็นวงจรอิเล็กทรอนิกส์ที่ออกแบบขึ้นมาใหม่โดยอ้างอิงหลักการเหนี่ยวนำด้วยพัลส์และวิธีการใช้งานในทางปฏิบัติ ดังแสดงไว้ในรูปที่ 2.16

¹ J. A. Corbyn. (1980). Pulse Induction Metal Detector – 1 and 2. Wireless World.

² DeepTech Metal Detectors. (2007). Coil and Search Head Design – Patents and Utility Models (Part 1). from

www.deeptech-bg.com/search_coils.pdf.



รูปที่ 2.16 วงจรเปิด/ปิดพัลส์ให้แก่ขดลวด (Coil Driver)

หลักการการทำงานของวงจรมีดังนี้

1. เริ่มต้น ไมโครคอนโทรลเลอร์ AVR ATmega168 จะสร้างสัญญาณ PWM โดยใช้ไทเมอร์ 0 ไปกระตุ้นให้ MOSFET IRFP630B ทำงานในสภาวะปิดวงจร ส่งผลให้กระแสไฟฟ้าจากแหล่งกำเนิดไฟฟ้าจะไหลผ่านวงจรให้แก่ขดลวดเพิ่มขึ้นเรื่อยๆ ในช่วงระยะเวลาหนึ่ง
2. เมื่อไทเมอร์ 0 สร้างพัลส์ของสัญญาณ PWM ด้านบวกเสร็จสิ้นตามค่าความกว้างพัลส์บวกที่กำหนด ใน ณ ขณะนั้น MOSFET จะทำงานในสภาวะเปิดวงจร นั่นคือ กระแสไฟฟ้าที่ไหลในขดลวดหรือวงจรจะเริ่มลดลงไปเรื่อยๆ จนกระทั่งกระแสไฟฟ้าที่ไหลในขดลวดมีค่าเท่ากับ 0
3. ในขณะที่กระแสไฟฟ้าที่ไหลในขดลวดเริ่มมีค่าลดลง AVR ATmega168 จะเริ่มต้นสั่งการให้ไทเมอร์ 1 ทำการจับระยะเวลาการลดลงของกระแสไฟฟ้าที่ไหลในขดลวด ซึ่งเริ่มจับเวลาตั้งแต่กระแสไฟฟ้าที่ไหลในขดลวดเริ่มมีค่าลดลงไปจนถึงเวลาที่กระแสไฟฟ้ามามีค่าเท่ากับ 0
4. เมื่อกระแสไฟฟ้าที่ไหลในขดลวดมีค่าเท่ากับ 0 จะทำให้เอาต์พุตของวงจรที่ต่ออยู่กับขาพอร์ต์อินเตอร์รัปต์ภายนอกช่องที่ 0 ของไมโครคอนโทรลเลอร์ทำงาน (ในที่นี้ กำหนดโหมดทำงานของสัญญาณอินเตอร์รัปต์เป็นแบบขอบขาลง) ซึ่งจะไปสั่งการให้ไทเมอร์ 1 หยุดจับเวลา
5. หลังจากนั้น AVR ATmega168 จะทำการบันทึกค่าระยะเวลาการลดลงของกระแสไฟฟ้าที่ไหลในขดลวด โดยในที่นี้ ค่าระยะเวลาที่ตรวจจับได้จะเป็นจำนวนคล็อกที่นับได้ในช่วงเวลาดังกล่าวในหน่วยเวลาของไทเมอร์ 1 ซึ่งค่าต่างๆ ที่ใช้ในการคำนวณและขั้นตอนวิธี (อัลกอริทึม) ในการเขียนโปรแกรม

ใช้งานไทมเมอร์ 0 และไทมเมอร์ 1 ของไมโครคอนโทรลเลอร์ AVR ATmega168 จะอธิบายไว้ในหัวข้อย่อยในบทที่ 3 ต่อไป

2.7 ผลงานวิจัยที่เกี่ยวข้อง

การออกแบบเครื่องตรวจจับโลหะโดยใช้หลักการเหนี่ยวนำด้วยพัลส์ (Pulse Induction <PI> Metal Detector) นั้นได้ถูกกล่าวถึงครั้งแรกในหนังสือต่างประเทศที่แต่งขึ้นโดย I. Grant, F.S. และ West, G.F.¹ ต่อมา J. A. Corbyn² ได้พัฒนาและนำเสนอวงจรตรวจจับโลหะขึ้นมาใหม่ ซึ่งประกอบไปด้วยตัว R ตัว ,C และทรานซิสเตอร์เป็นจำนวนมาก ทำให้การทดสอบคุณสมบัติในการใช้งานนั้นกระทำได้ยากมาก ต้องเสียเวลาปรับจูนหลายครั้ง ก่อนที่จะใช้งานได้จริงและมีคุณสมบัติใช้งานที่เป็นไปตามความต้องการ ต่อจากนั้น ยังคงมีนักวิจัยและบริษัทต่างๆ หลายประเทศจำนวนมากได้ทำการวิจัย พัฒนา รวมไปถึงสร้างเพื่อขายเครื่องตรวจจับโลหะออกอย่างต่อเนื่อง ซึ่งโดยส่วนใหญ่จะเน้นไปที่การสร้างเพื่อขายและเน้นจดสิทธิบัตรเพื่อขายลิขสิทธิ์ในการสร้างเครื่องตรวจจับโลหะ โดยในที่นี่ จะขอกล่าวถึงบทความวิจัยที่สำคัญที่นำมาใช้ในงานวิจัยฉบับนี้ ดังนี้

Dr. Salman *et. al.*³ ได้นำเสนอเครื่องตรวจจับโลหะแบบการเหนี่ยวนำด้วยพัลส์ขึ้นมา โดยเครื่องดังกล่าวประกอบไปด้วยวงจรที่ใช้ในการสร้างพัลส์ที่มีค่าความถี่และความกว้างของพัลส์คงที่โดยใช้ไอซีไทมเมอร์ LM555 4 ตัว, วงจรที่ใช้ในการตรวจจับโลหะโดยใช้วงจรขยายสัญญาณและวงจรอินทิเกรเตอร์, แหล่งกำเนิดไฟฟ้า +5V, -5V และ -10V และขดลวดที่ใช้ตรวจจับ 1 วง โดยค่าความถี่และความกว้างของพัลส์สามารถที่จะเปลี่ยนแปลงค่าได้โดยการปรับจูนค่าความต้านทานและค่าตัวเก็บประจุ หลักการทำงานคือ หลังจากที่พัลส์ถูกส่งออกไปภาคส่งแล้ว วงจรซิกตัวอย่าง (Sampling Circuit) ที่อยู่ในวงจรตรวจจับโลหะจะทำการมอนิเตอร์ตรวจจับระยะเวลาของพัลส์ที่สะท้อนกลับมา (Reflected Pulse หรือ Echo) ซึ่งในสภาวะปกติที่ไม่มีโลหะ ระยะเวลาการลดลง (Delay) ของพัลส์ที่สะท้อนกลับมาและตรวจจับได้ จะถูกกำหนดเป็นค่าระยะเวลาการลดลงขีดเริ่ม (Expected Length) ซึ่งมีค่าประมาณ 30 μ s แต่ถ้ามีการตรวจจับพบโลหะ ระยะเวลาการลดลงของพัลส์ที่สะท้อนกลับมาและตรวจจับได้จะมีค่าที่มากกว่าค่าระยะเวลาการลดลงขีดเริ่มประมาณ 2 – 3 μ s โดยได้สรุปประเด็นสำคัญที่นำเสนอไว้ว่า ถึงแม้ว่า PI Metal Detector จะไม่สามารถใช้งานในการจำแนกชนิดของโลหะได้ แต่ก็มีข้อดีคือสามารถตรวจจับโลหะภายใต้สภาพแวดล้อม

¹ I. Grant, F. S. and West, G. F. (1965). **Interpretation Theory in Applied Geophysics**. McGraw-Hill.

² J. A. Corbyn. (1980). Pulse Induction Metal Detector – 1 and 2. **Wireless World**.

³ Salman Durrani, James Boxall, Stephen Purvis, Garrick Madge, and Tathagat Banerjee. A Pulse Induction Metal Detector (ENGN3227). from http://users.cecs.anu.edu.au/~Salman.Durrani/_teaching/TA5.pdf.

หรือพื้นที่ที่มีค่าความนำไฟฟ้าสูงและสามารถตรวจจับโลหะที่อยู่ลึกกว่าเครื่องตรวจจับโลหะชนิดอื่นๆ โดยในงานวิจัยนี้ไม่ได้แสดงผลการทดสอบที่เกิดขึ้นจากการตรวจจับเลย นอกจากนี้ ข้อเสียของ PI Metal Detector ที่พบจากการนำเสนอดังกล่าวคือ เครื่องตรวจจับโลหะดังกล่าวประกอบไปด้วยอุปกรณ์ RLC และไอซีจำนวนมาก ซึ่งจะเสียเวลาในการปรับจูนเวลานาน จนกระทั่งได้ค่าความถี่และความกว้างพัลส์ตามที่ต้องการ, ต้องใช้ไฟเลี้ยงจำนวน 3 ค่า ซึ่งเป็นการยุ่งยากในการทำงาน อีกทั้งไม่สามารถแยกแยะโลหะที่ต้องการจะตรวจจับได้

Nina Gajjar ¹ ได้นำเสนอเครื่องตรวจจับโลหะอย่างง่ายที่ใช้อุปกรณ์ RLC ไอซีขยายสัญญาณเสียงออกดีโอ ร่วมกับสายอากาศแบบลูป (Loop Antenna) 2 ตัว โดยเครื่องตรวจจับโลหะนี้จะประกอบไปด้วย วงจรออสซิลเลเตอร์ 2 วงจรที่เป็นแบบปรับค่าความถี่ได้และเป็นแบบที่มีค่าความถี่ตายตัว สัญญาณเอาต์พุตของทั้งสองวงจรออสซิลเลเตอร์นี้จะถูกนำมาผสมกันด้วยวงจร Mixer ที่ซึ่งจะสร้างสัญญาณผลต่างออกมาเท่านั้น และในตอนแรก ทั้งสองวงจรถูกปรับจูนให้มีความถี่ตรงกันเสียก่อน (Zero Beat) โดยใช้วงจรควบคุมการปรับจูนความถี่และจะต้องปรับจูนในสภาวะที่ไม่มีโลหะ ซึ่งจะไม่มีการเกิดเสียงออกดีโอเกิดขึ้น แต่ถ้านำไปทดสอบและตรวจจับพบโลหะ ค่า Zero Beat จะไม่เกิดขึ้นและมีเสียงออกดีโอเกิดขึ้นที่ลำโพง โดยผลการทดสอบที่เกิดขึ้นจากสิ่งประดิษฐ์นี้ก็เป็นเพียงแต่การตรวจจับพบกับไม่พบโลหะ และมีเสียงเท่านั้น ไม่สามารถแยกชนิดของโลหะที่ตรวจพบได้

Michael R. Starcher ² ได้นำเสนอ Blind Squirrel PI Metal Detector ที่ซึ่งเครื่องตรวจจับโลหะที่นำเสนอนี้ประกอบไปด้วยอุปกรณ์ RLC, MOSFET และไอซีจำนวนมาก ได้แก่ LM318, LM555, 4N35 ฯลฯ และขดลวดตรวจจับ 1 วง โดยพัลส์ที่มีช่วงเวลา ON สั้นๆ มีความถี่คงที่และมีค่ากระแสไฟฟ้าสูงได้ถูกสร้างขึ้นตลอดเวลาและส่งไปยังขดลวดโดยใช้ไอซีไทมเมอร์ 555 เช่นเดียวกับงานวิจัยของ Dr. Salman *et. al.* แต่ใช้งานวงจรอิเล็กทรอนิกส์ที่แตกต่างกัน จึงทำให้ความถี่ใช้งานมีค่าไม่เท่ากันด้วย โดยค่าความถี่ที่ใช้งานจะมีค่าอยู่ระหว่าง 75 Hz และ 225 Hz ด้วยค่าความกว้างพัลส์ที่มีค่าประมาณ 100 - 175 μ s เส้นผ่านศูนย์กลางด้านในและด้านนอกของขดลวดที่ใช้ในการตรวจจับโลหะมีค่าเท่ากับ 6.5 นิ้ว (16.51 ซม.) และ 9.5 นิ้ว (24.13 ซม.) ตามลำดับ โดยงานวิจัยที่นำเสนอนี้ก็ไม่ได้แสดงผลการทดสอบที่เกิดขึ้นจากการตรวจจับเช่นกัน แต่เข้าใจว่า เครื่องดังกล่าวสามารถตรวจจับโลหะ แต่ไม่สามารถแยกชนิดของโลหะที่ตรวจพบได้เช่นกัน

¹Nina Gajjar. FET Metal detector with IC Audio Amplifier. from <http://nina.foxdelta.com/detector.html>.

²Michael R. Starcher. Blind Squirrel PI Metal Detector. from <http://oldradiobuilder.com/MDET.html>.



นอกจากนี้ จากการศึกษาและรวบรวมข้อมูลเกี่ยวกับเครื่องตรวจจับ โลหะที่มีขายในประเทศไทย พบว่า เครื่องตรวจจับโลหะส่วนใหญ่จะนำเข้ามาจากต่างประเทศ ซึ่งมีอยู่หลายบริษัทและหลายรุ่น ขึ้นอยู่กับสภาพแวดล้อมของการทำงานและคุณสมบัติในการทำงาน ยกตัวอย่างเช่น เครื่องตรวจจับโลหะแบบพกพา รุ่น MD-1001¹ เป็นเครื่องตรวจจับโลหะที่สามารถพกพาติดตัวได้เพื่อใช้ในการตรวจจับโลหะในงานรักษาความปลอดภัยตามสถานที่ที่ต้องการปลอดจากอาวุธ อาทิ สถานที่ราชการ สนามบิน สถานีขนส่ง ฯลฯ ซึ่งมีคุณสมบัติในการทำงานคือ สามารถปรับความไวในการตรวจจับโลหะ เพื่อให้เหมาะสมกับชนิดของโลหะเป้าหมายได้, มีสวิทช์เปิด/ปิดเพื่อการประหยัดแบตเตอรี่ ในกรณีที่พักการใช้งาน และมีเสียงเตือนเมื่อมีการตรวจจับโลหะได้ หลักการทำงานของ Handheld Metal Detector รุ่นนี้ก็คือ ตัวเครื่องจะให้กำเนิดสนามแม่เหล็กออกมา เมื่อมีโลหะที่เป็น Ferrite มาตัดผ่านก็จะเกิดการเหนี่ยวนำให้เกิดกระแสไฟฟ้าค่าน้อยๆ ขึ้นในวงจรตรวจจับ แล้วส่งเสียงเตือนออกมา โดยปัจจัยที่จะส่งผลต่อการตรวจจับโลหะนั้นขึ้นอยู่กับชนิดของโลหะ, ขนาดของโลหะ และความห่างจากผิวหน้าของเครื่อง

เครื่องตรวจจับวัตถุโลหะรุ่น Garrett Super Scanner² สามารถตรวจจับวัตถุจำพวก ปืนพกขนาดกลาง, มีดขนาดใหญ่, มีดคัตเตอร์, ฟอยล์ห่อยาเสพติด/เพชรเม็ดเล็กๆ ได้ในระยะทาง 9 นิ้ว, 6 นิ้ว, 3 นิ้ว และ 1 นิ้ว ตามลำดับ ซึ่งโดยทั่วไปจะใช้งานในอากาศเท่านั้น ยังไม่ทราบผลการตรวจจับโลหะที่วางในพื้นที่ดิน ซึ่งตรงข้ามกับเครื่องตรวจจับโลหะที่ใช้งานในทางทหาร VMH3CS Mine Detector รุ่น VMW1 กับ VMW3 ของบริษัท Vallon³ ที่นำเข้ามาจากประเทศเยอรมนี เพื่อใช้ในการตรวจจับกับดักระเบิดและวัตถุต้องสงสัยที่เป็นโลหะต่างๆ ที่วางอยู่ในพื้นดิน

ดังนั้น จึงเกิดแนวความคิดที่จะทำการศึกษาและออกแบบวงจรอิเล็กทรอนิกส์ใหม่เพื่อทำหน้าที่เปิด/ปิดวงจรพัลส์ให้แก่เครื่องตรวจจับโลหะแบบที่ใช้การเหนี่ยวนำด้วยพัลส์และเขียนโปรแกรม ARM LPC17687 ขึ้นมาเพื่อสร้างสัญญาณพัลส์ PWM แทนการใช้อุปกรณ์ RLC และไอซีจำนวนมากที่มีการนำเสนอในงานวิจัยก่อนหน้านี้ รวมถึงจัดเก็บค่าระยะเวลาการลดลงของกระแสไฟฟ้าที่ไหลในขดลวดที่เกิดจากโลหะแต่ละชนิด โดยจะทำการตรวจจับโลหะทั้งที่วางในอากาศและที่วางในพื้นที่ดิน รวมถึงจะทำการเปรียบเทียบระยะทางที่ตรวจจับโลหะได้กับระยะทางที่เครื่องตรวจจับวัตถุโลหะรุ่น Garrett Super Scanner

¹ Handheld Metal Detector (MD-1001). from

<http://www.gunkamoy.com/index.php?lay=show&ac=article&Id=352431&Ntype=4>.

² Garrett Super Scanner (PI Metal Detector). from

http://www.garrett.com/security/s_superscanner_specification.pdf.

³ VMH3CS Mine Detector. from http://www.vallon.de/pdf/VMH3CS_leaflet_09_2010.pdf.