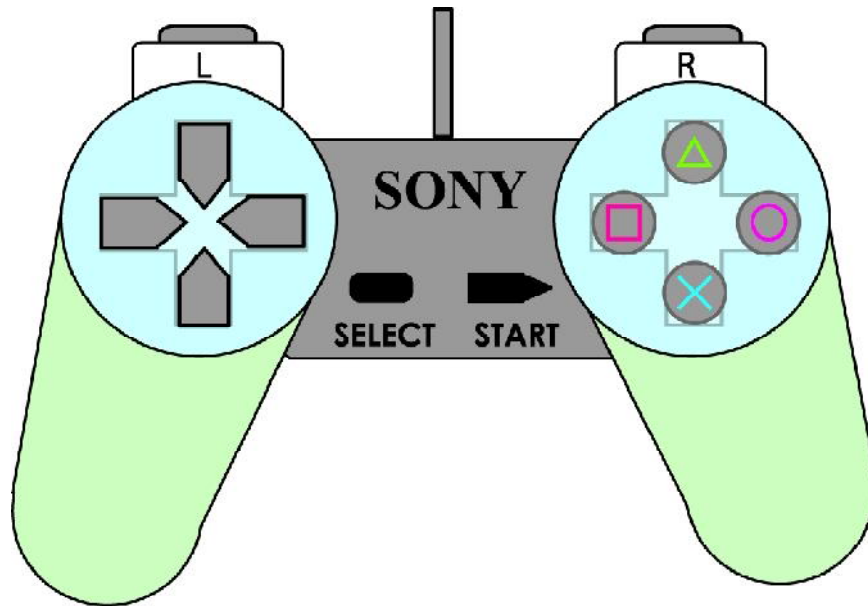


# การเชื่อมต่อจอยสติ๊ก PS1 เข้ากับบอร์ด RBX-877

จอยสติ๊ก PS1 ปกติจะใช้กับเครื่องเล่นเกมเพลย์สเตชัน ของบริษัทโซนี่ เป็นเครื่องเล่นเกมที่ได้รับความนิยมสูงมาก ดังนั้นอุปกรณ์ต่อพ่วงต่าง ๆ จึงหาซื้อได้ง่ายและราคาไม่แพง นอกจากนี้การเขียนโปรแกรมเพื่อติดต่อยังทำได้ง่ายอีกด้วย

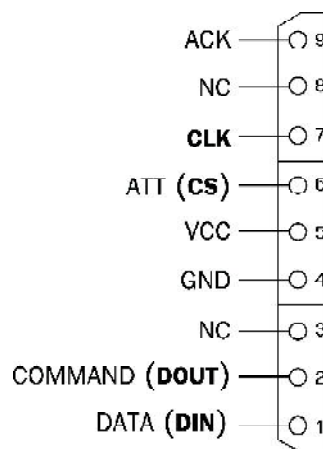
## ลักษณะของจอยสติ๊ก PS1 และขาเชื่อมต่อ

จอยสติ๊ก PS1 มีลักษณะดังแสดงในรูปที่ 1 มีปุ่มกดทั้งหมด 14 ปุ่มด้วยกัน



### รูปที่ 1 ลักษณะหน้าตาของจอยสติ๊ก PS1

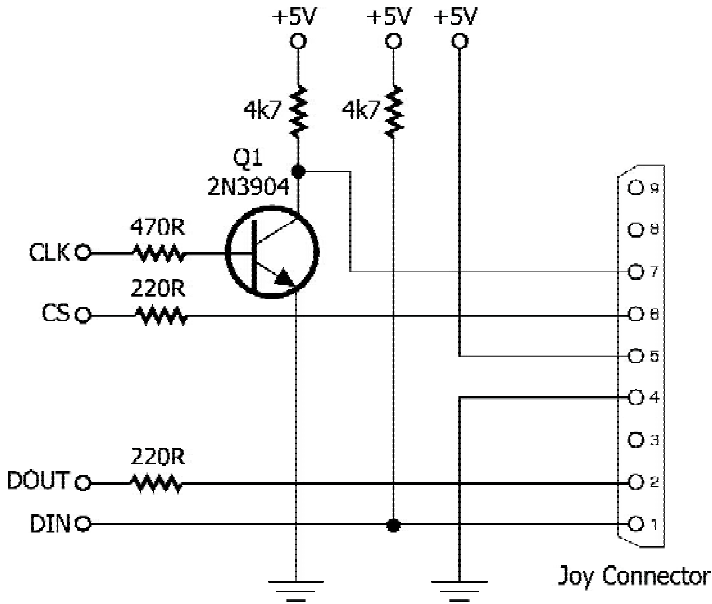
สำหรับคอนเน็กเตอร์ที่ใช้เชื่อมต่อจะมีลักษณะพิเศษตามมาตรฐานของโซนี่ ดังแสดงในรูปที่ 2 โดยจะมี จุดต่อทั้งหมด 9 ขา แต่จะมีขาที่ใช้งานจริง ๆ เพียงแค่ 4 ขาเท่านั้น คือ DIN ,DOUT ,CLK และ CS ซึ่งลักษณะการเชื่อมต่อกับไมโครคอนโทรลเลอร์จะใช้การสื่อสารอนุกรมตามมาตรฐาน SPI



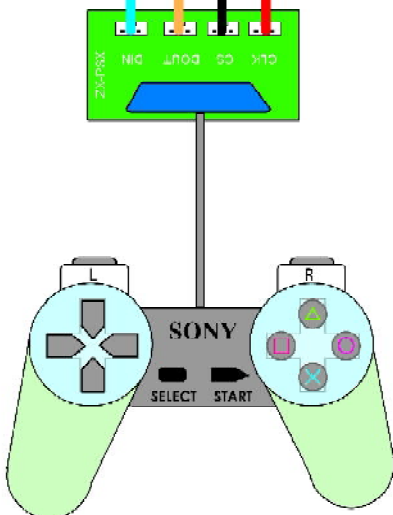
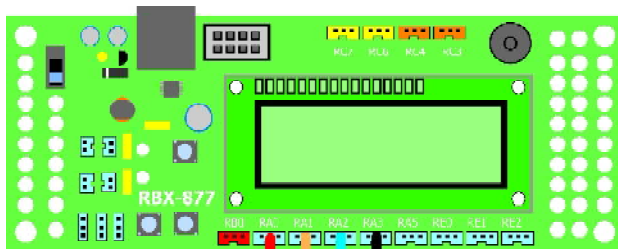
Joy PSX Connector

### รูปที่ 2 คอนเน็กเตอร์ของจอยสติ๊ก PS1

โดยจะต้องต่อวงจรเพิ่มเติมบางส่วนเพื่อให้สามารถเชื่อมต่อเข้ากับไมโครคอนโทรลเลอร์ได้โดยตรง โดยลักษณะวงจรที่ใช้เชื่อมต่อแสดงในรูปที่ 3 (ก) จากวงจรมานำมาสร้างเป็นบอร์ด ZX-PSX ดังแสดงในรูปที่ 3(ข) ซึ่งสามารถต่อพ่วงบอร์ดนี้เข้ากับ บอร์ดไมโครคอนโทรลเลอร์ต่าง ๆ ได้โดยตรงตัวอย่างการเชื่อมต่อเข้ากับบอร์ด RBX-877 ดังแสดงรูปแบบการเชื่อมต่อในรูปที่ 4



รูปที่ 3 วงจรเชื่อมต่ออินเทอร์เฟซของจอยสติ๊ก PSX กับไมโครคอนโทรลเลอร์ และแผงวงจร ZX-PSX ที่ประกอบเรียบร้อยแล้ว



RA0 เชื่อมต่อกับขา CLK  
 RA1 เชื่อมต่อกับขา DOUT  
 RA2 เชื่อมต่อกับขา DIN  
 RA3 เชื่อมต่อกับขา CS

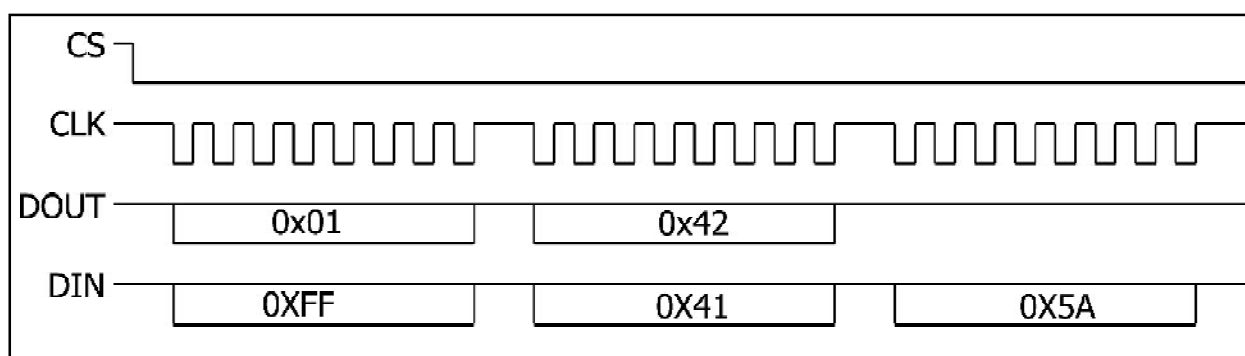
รูปที่ 4 แสดงตัวอย่างการเชื่อมต่อระหว่างบอร์ด RBX-877 กับจอยสติ๊ก PSX

## การเขียนโปรแกรมเพื่อติดต่อกับจอยสติ๊ก PS1

การสื่อสารจอยสติ๊ก PS1 จะใช้รูปแบบการสื่อสารอนุกรมแบบซิงโครนัส (มีสัญญาณนาฬิกาเป็นตัวกำกับ) โดยมีขั้นตอนการติดต่อดังนี้

1. ให้ขา CS (Chip Select) มีลอจิก "0" เพื่อเลือกจอยสติ๊กที่ต้องการติดต่อด้วย
2. ส่งข้อมูล 0x01 พร้อม ๆ กับสัญญาณนาฬิกาเพื่อแจ้งการเริ่มต้นการติดต่อ
3. ส่งข้อมูล 0x42 เพื่ออ่านขออ่านค่าการกดคีย์จากจอยสติ๊ก PS1 พร้อม ๆ กันนี้จอยสติ๊กจะส่งค่า ID ชนิดของจอยที่ใช้เชื่อมต่อ โดยจอยสติ๊ก PS1 จะมีค่า ID เท่ากับ 0x41
4. ในสัญญาณชุดถัดมา จอยสติ๊ก PS1 จะส่ง 0x5A ตอบกลับมาจากนั้นจะสามารถอ่านข้อมูลคีย์ต่าง ๆ ได้ในสัญญาณนาฬิกาชุดถัดไป

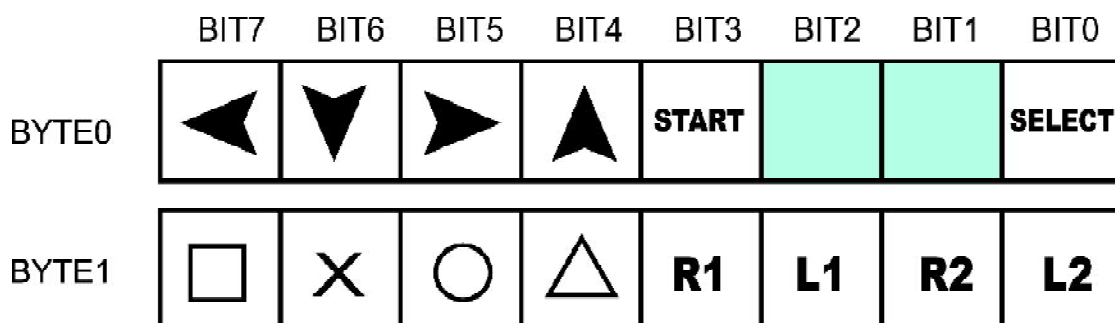
รายละเอียด Timing Diagram แสดงในรูปที่ 5



รูปที่ 5 ไทมิ่งไดอะแกรมแสดงสัญญาณที่ใช้ติดต่อกับจอยสติ๊ก PSX

## ตำแหน่งปุ่มกดของจอยสติ๊ก PS1 เมื่อเทียบกับข้อมูลที่อ่านได้

สำหรับจอยสติ๊ก PS1 มาตรฐาน จะมีไบต์ที่ต้องอ่านค่าเพียงสองไบต์ เป็นข้อมูลต่อจากข้อมูล 0x5A ก่อนหน้านี้ โดยตำแหน่งบิตต่าง ๆ แสดงในรูปที่ 6 เมื่อกดที่คีย์ใดคีย์หนึ่งค่าจะเปลี่ยนจากลอจิก "1" เป็นลอจิก "0" เฉพาะที่บิตนั้น ๆ



รูปที่ 6 ตำแหน่งบิตต่าง ๆ ของสวิทช์บนจอยสติ๊ก PS1

จากรูปแบบข้างต้นสามารถนำมาเขียนเป็นโปรแกรม เพื่อนำค่าที่อ่านได้แสดงผลที่หน้าจอ LCD ของบอร์ด RBX-877 ดังแสดงในโปรแกรมที่ 1

```

/*****Define Bit for Connect ZX-PSX*****/
#define CLK      PortA.F0
#define DOUT     PortA.F1
#define DIN      PortA.F2
#define CS       PortA.F3

unsigned char PSX_IN;
unsigned char PSX_ID;
unsigned char i;
unsigned char PSX_STATUS;
unsigned char PSX_Left;
unsigned char PSX_RIGHT;
unsigned char dec_[4];

unsigned char PSX_TxRx(unsigned char BYTE_OUT){
    unsigned char BYTE_IN=0;
    unsigned char j=0x01;
    for (i=0;i<8;i++){
        if ((BYTE_OUT&j)==j ) // Loop 8 Time for Send and Recieve Byte
            { DOUT=1;} // Test Bit 0 or 1
        else // Set Dout
            { DOUT=0;}
        CLK=1; // Clock Pulse High
        BYTE_IN=(BYTE_IN>>1); // Receive Byte LSB First
        if (DIN==1){BYTE_IN=(BYTE_IN | 0x80);}
        CLK=0; // Clock Pulse Low
        j=j<<1;
    }
    return BYTE_IN;
}

void main(){
    ANSEL=0x00; // Set PORTA to digital I/O
    TRISA=0b00000100;
    Lcd_Init(&PORTD); // Initial LCD
    Lcd_Cmd(LCD_CURSOR_OFF);
    CLK=0;
    CS=1;
    DOUT=0;
    Delay_ms(500);
    while(1){
        CS=0;
        PSX_IN=PSX_TxRx(0x01); // Send Start Command
        PSX_ID=PSX_TxRx(0x42); // Send Command Read Switch and Read ID
        PSX_STATUS=PSX_TxRx(0x00); // Read STATUS (0x5A)
        PSX_LEFT=PSX_TxRx(0x00); // Read Byte 0
        PSX_RIGHT=PSX_TxRx(0x00); // Read Byte 1
        CS=1;
        /***** Show Data On LCD *****/
        BytetoStr(PSX_ID,dec_);
        Lcd_Out(1,1,dec_);
        BytetoStr(PSX_STATUS,dec_);
        Lcd_Out(1,5,dec_);
        BytetoStr(PSX_LEFT,dec_);
        Lcd_Out(1,9,dec_);
        BytetoStr(PSX_RIGHT,dec_);
        Lcd_Out(1,13,dec_);
    }
}

```