

**PRESENTER: 郭政穎**

# **ADSP FINAL PROJECT**

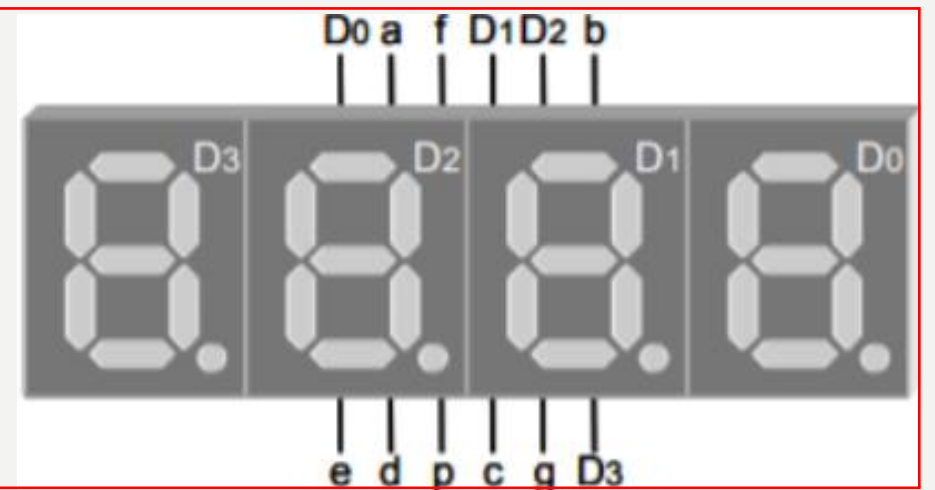
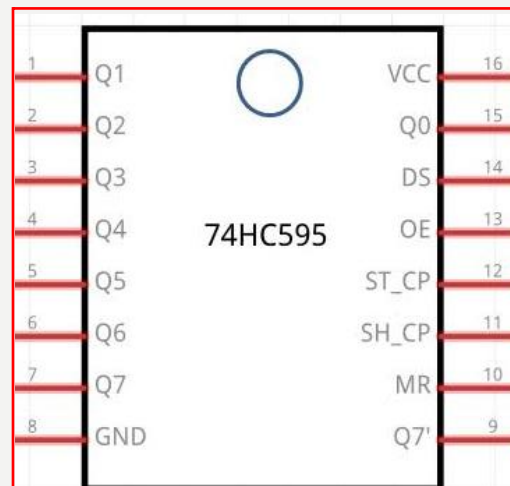
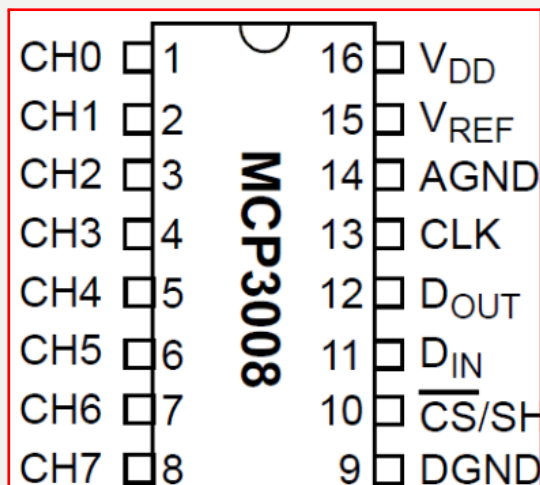
**SIGNAL PROCESSING ON AN  
EMBEDDED SYSTEM**

# OUTLINE

- I/O devices components
- Code review
- Perf observation
- Application
- Reference

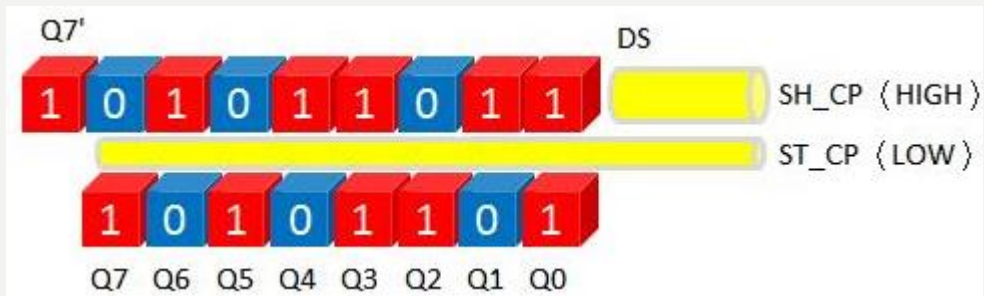
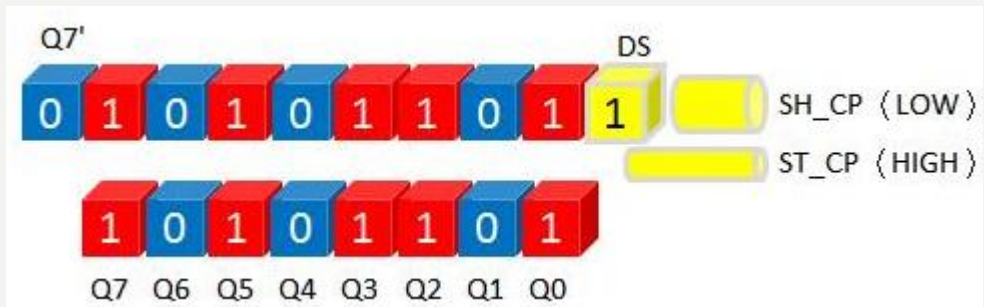
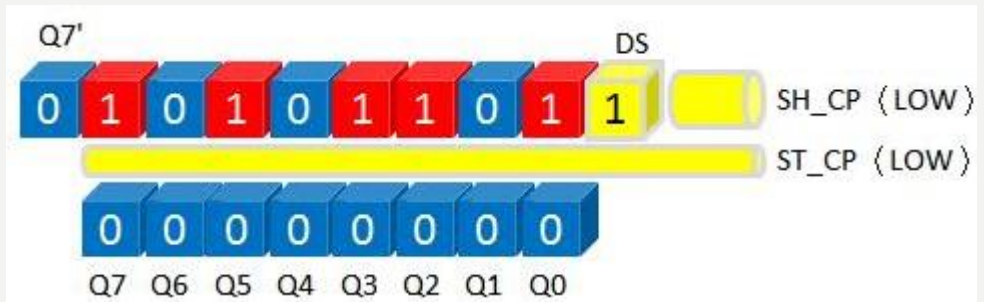
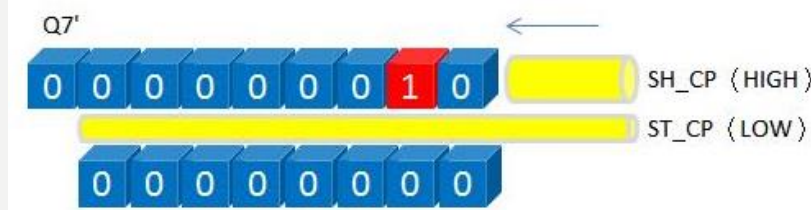
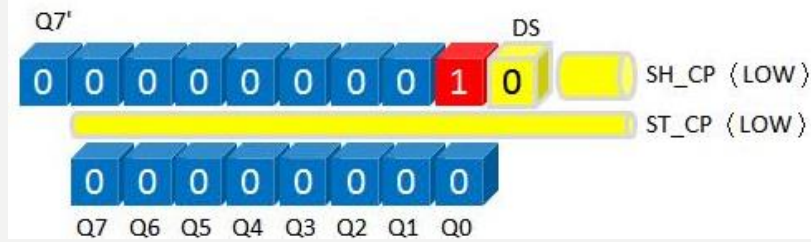
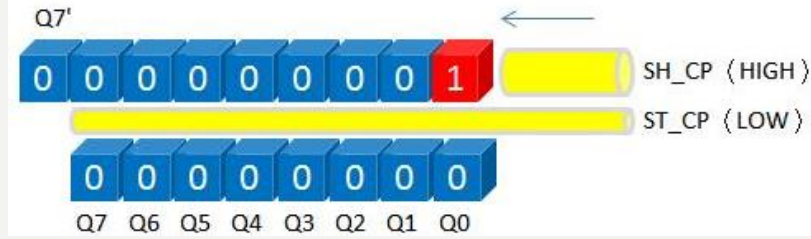
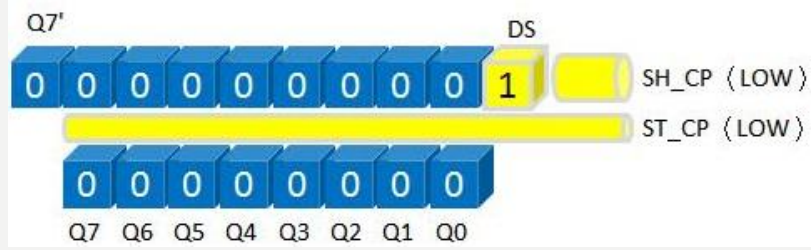
# I/O DEVICES COMPONENTS

- Controller Board: Raspberry Pi
- microphone module: KY-038
- Integrated circuits: **MCP3008(ADC)** 、74hc595(Shift register)
- 4-Digit 7-Segment display
- Others: Button 、Resistance\*8 、DuPont wire(Jumper Wire)\*數條

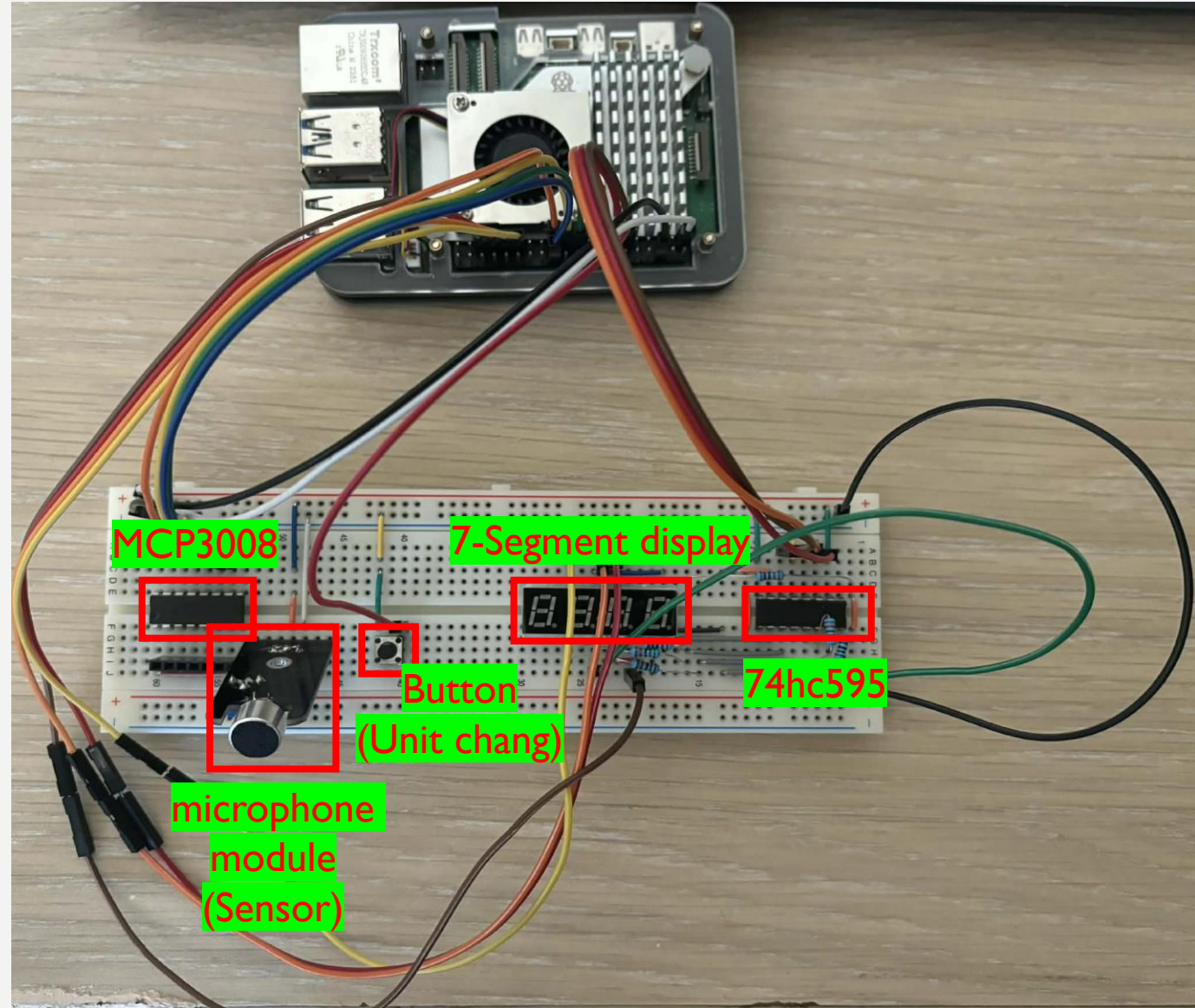
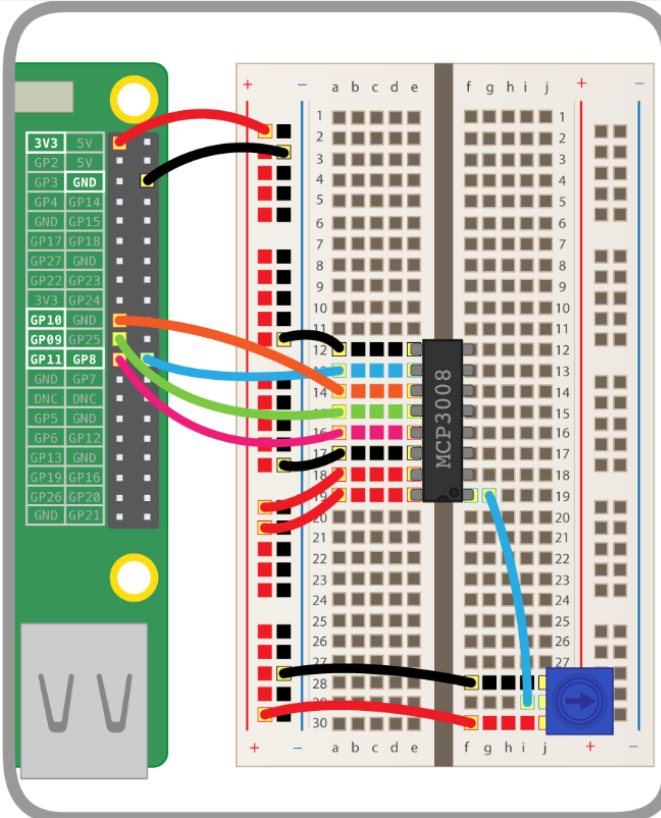
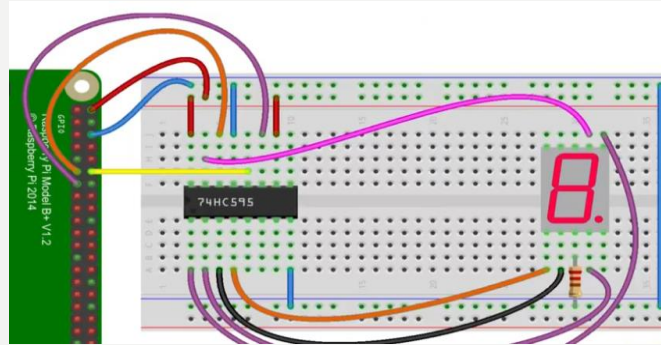




# 74HC595



# SIMULATION AND ENTITY WIRING



# CODE REVIEW-KERNEL DRIVER

```
23  /* Read MCP3008 CH0~CH7 */
24  static int mcp3008_read_channel(struct spi_device *spi, u8 ch)
25  {
26      u8 tx[3];
27      u8 rx[3];
28      struct spi_transfer t = {
29          .tx_buf = tx,
30          .rx_buf = rx,
31          .len = 3,
32          .speed_hz = 1350000,
33          .bits_per_word = 8,
34      };
35      struct spi_message m;
36      int ret;
37
38      if (ch > 7)
39          return -EINVAL;
40
41      tx[0] = 0x01;
42      tx[1] = 0x80 | (ch << 4);
43      tx[2] = 0x00;
44
45      spi_message_init(&m);
46      spi_message_add_tail(&t, &m);
47
48      ret = spi_sync(spi, &m);
49      if (ret < 0)
50          return ret;
51
52      return ((rx[1] & 0x03) << 8) | rx[2];
53 }
```

```
static ssize_t mcp3008_read(struct file *f, char __user *buf, size_t len, loff_t *off)
{
    int adc;
    char tmp[16];
    int n;

    mutex_lock(&mcp.lock);
    adc = mcp3008_read_channel(mcp.spi, 0);
    mutex_unlock(&mcp.lock);
    n = snprintf(tmp, sizeof(tmp), "%d\n", adc);
    if (n < 0)
        return n;

    if (len < n)
        return -EINVAL;

    if (copy_to_user(buf, tmp, n))
        return -EFAULT;

    return n;
}
```

# CODE REVIEW-USER SPACE

```
57  /* =====
58  * ADC
59  * ===== */
60  static int read_adc_from_dev(void)
61  {
62      int fd = open("/dev/mcp3008", O_RDONLY);
63      if (fd < 0) return -1;
64
65      char buf[32];
66      int ret = read(fd, buf, sizeof(buf) - 1);
67      close(fd);
68
69      if (ret <= 0) return -1;
70      buf[ret] = '\0';
71      return atoi(buf);
72  }
```

```
/* =====
 * ADC → SPL (instant)
 * ===== */
static int adc_to_spl(int adc)
{
    double v = (double)adc * VREF / 1023.0;
    double p = v / (MIC_SEN_V_PER_PA * PREAMP_GAIN);

    double spl = 20.0 * log10(fmax(p, 1e-12) / P_REF);

    if (spl < 0) spl = 0;
    if (spl > 999) spl = 999;

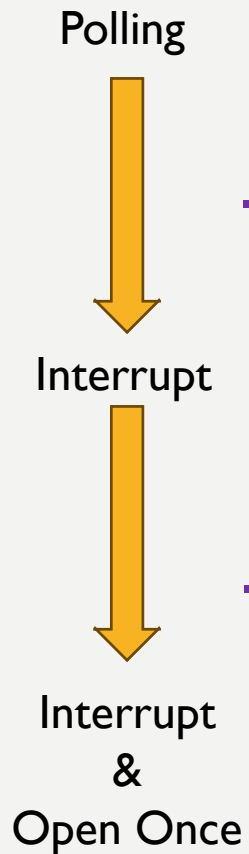
    return (int)lround(spl);
}
```

# CODE REVIEW-USER SPACE (OPEN ONCE)

```
90  /* =====  
91  * ADC read (open once)  
92  * ===== */  
93  static int read_adc_from_dev(void)  
94  {  
95      char buf[32];  
96  
97      lseek adc_fd, 0, SEEK_SET;  
98  
99      int ret = read(adc_fd, buf, sizeof(buf) - 1);  
100     if (ret <= 0)  
101         return -1;  
102  
103     buf[ret] = '\0';  
104     return atoi(buf);  
105 }
```

```
int main(void)  
{  
    if (gpio_init() < 0) {  
        fprintf(stderr, "GPIO init failed\n");  
        return 1;  
    }  
  
    adc_fd = open("/dev/mcp3008", O_RDONLY);  
    if (adc_fd < 0) {  
        perror("open /dev/mcp3008");  
        return 1;  
    }  
  
    printf("MODE 0=V | 1=ADC | 2=dB\n");  
  
    while (1) {  
        check_button();  
  
        int adc = read_adc_from_dev();  
        if (adc < 0) continue;  
  
        double v = adc * VREF / 1023.0;  
        int spl = adc_to_spl(adc);  
  
        char disp[8];  
        if (MODE == 0)  
            snprintf(disp, sizeof(disp), "%.2f", v);  
        else if (MODE == 1)  
            snprintf(disp, sizeof(disp), "%4d", adc);  
        else  
            snprintf(disp, sizeof(disp), "%4d", spl);  
  
        printf("\rMODE=%d ADC=%4d V=%4.2f SPL=%3d dB DISP='%s' ",  
            MODE, adc, v, spl, disp);  
        fflush(stdout);  
  
        for (int i = 0; i < 80; i++)  
            display_number(disp);  
    }  
}
```

# PERF OBSERVATION



```
James@james:~/mcp3008_driver$ sudo perf stat --timeout 10000 -e instructions,cycles,cache-references,cache-misses,branch-misses ./decibel
MODE=0 V | MODE=1 ADC | MODE=2 dB
MODE=1 ADC= 7 V=0.02 SPL= 87 dB DISP=' 7' ./decibel: Terminated

Performance counter stats for './decibel':

   213,755,218   instructions          #    0.88 insn per cycle
   243,806,448   cycles
   52,976,296    cache-references
   196,755      cache-misses          #    0.37% of all cache refs
   485,192      branch-misses

10.001325252 seconds time elapsed

0.013736000 seconds user
0.154690000 seconds sys
```

```
James@james:~/mcp3008_driver$ sudo perf stat --timeout 10000 -e instructions,cycles,cache-references,cache-misses,branch-misses ./decibel_interrupt
MODE 0=V | 1=ADC | 2=dB
MODE=0 ADC= 0 V=0.00 SPL= 0 dB DISP='0.00'
MODE → 1
MODE=1 ADC= 0 V=0.00 SPL= 0 dB DISP=' 0' ./decibel_interrupt: Terminated

Performance counter stats for './decibel_interrupt':

   215,271,852   instructions          #    0.91 insn per cycle
   237,099,101   cycles
   53,016,241    cache-references
   176,400      cache-misses          #    0.33% of all cache refs
   470,181      branch-misses

10.010372457 seconds time elapsed

0.013612000 seconds user
0.163353000 seconds sys
```

```
James@james:~/mcp3008_driver$ sudo perf stat --timeout 10000 -e instructions,cycles,cache-references,cache-misses,branch-misses ./decibel_open_once
MODE 0=V | 1=ADC | 2=dB
MODE=1 ADC= 23 V=0.07 SPL= 97 dB DISP=' 23' ./decibel_open_once: Terminated

Performance counter stats for './decibel_open_once':

   213,722,625   instructions          #    0.91 insn per cycle
   235,495,988   cycles
   52,882,564    cache-references
   134,238      cache-misses          #    0.25% of all cache refs
   475,028      branch-misses

10.008954396 seconds time elapsed

0.021265000 seconds user
0.154177000 seconds sys
```

# APPLICATION AND DEMO

- The analog value read by the sensor(This uses a microphone module) is transmitted to the Raspberry Pi via the MCP3008 integrated circuit, and the output is displayed on a 7-segment display via the 74HC595 integrated circuit.The display unit can be changed by buttons.

**Mode 0** represents **voltage**, which will be displayed synchronously on the seven-segment display.  
The range in: 0V~3.3V



```
james@james:~/mcp3008_driver $ sudo ./decibel_open_once
MODE 0=V | 1=ADC | 2=dB
MODE=0 ADC= 455 V=1.47 SPL=123 dB DISP='1.47' |
```

**Mode 1** represents **ADC raw data**, which will be displayed synchronously on the seven-segment display.  
The range in: 0~1023



```
james@james:~/mcp3008_driver $ sudo ./decibel_open_once
MODE 0=V | 1=ADC | 2=dB
MODE=1 ADC= 186 V=0.60 SPL=116 dB DISP=' 186' |
```

**Mode 2** represents **decibel**, which will be displayed synchronously on the seven-segment display.  
The range in: 0dB~130dB



```
james@james:~/mcp3008_driver $ sudo ./decibel_open_once
MODE 0=V | 1=ADC | 2=dB
MODE=2 ADC= 9 V=0.03 SPL= 89 dB DISP=' 89' |
```

# REFERENCE

- <https://ricelee.com/product/MCP3008-IC>
- <https://openhome.cc/Gossip/CodeData/mBlockArduino/mBlockArduino18.html>
- <https://www.x-family.com.tw/?p=1384>
- <https://forums.raspberrypi.com/viewtopic.php?t=214202>
- <https://github.com/nagimov/mcp3008hwspi>
- [https://github.com/adafruit/Adafruit\\_CircuitPython\\_MCP3xxx/](https://github.com/adafruit/Adafruit_CircuitPython_MCP3xxx/)
- <https://forums.raspberrypi.com/viewtopic.php?t=192836>

**THANKS FOR YOUR LISTENING.**