

# การใช้งานไมโครคอนโทรลเลอร์ Arduino กับ Real-time clock DS1302

รศ.ณรงค์ บวบทอง

# DS1302

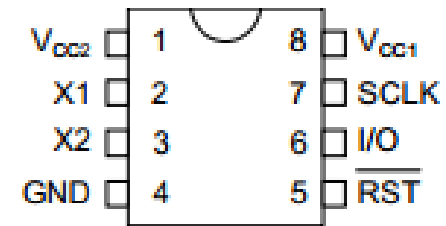
## Trickle Charge Timekeeping Chip

### FEATURES

- Real-time clock (RTC) counts seconds, minutes hours, date of the month, month, day of the week, and year with leap-year compensation valid up to 2100
- 31-byte, battery-backed, nonvolatile (NV) RAM for data storage
- Serial I/O for minimum pin count
- 2.0V to 5.5V full operation
- Uses less than 300nA at 2.0V
- Burst mode for reading/writing successive addresses in clock/RAM
- 8-pin DIP or optional 8-pin SOICs for surface Mount
- Simple 3-wire interface
- TTL-compatible (VCC = 5V)
- Optional industrial temperature range: -40°C to +85°C
- DS1202 compatible
- Underwriters Laboratory (UL) recognized

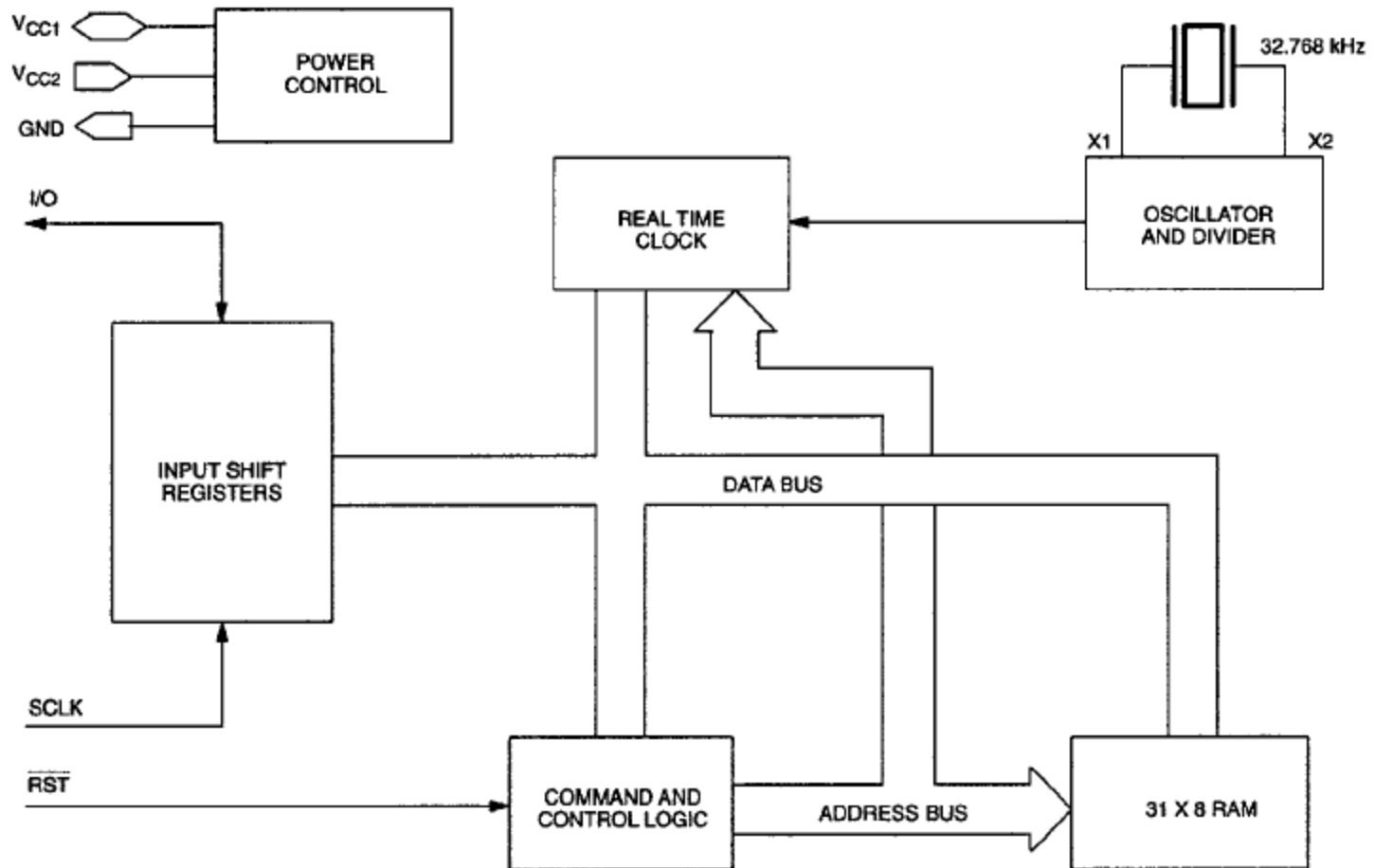
# DS1302 (ต่อ)

- X1, X2 - 32.768kHz Crystal Pins
- GND - Ground
- RST - Reset
- I/O - Data Input/Output
- SCLK - Serial Clock
- VCC1, VCC2 - Power Supply Pin



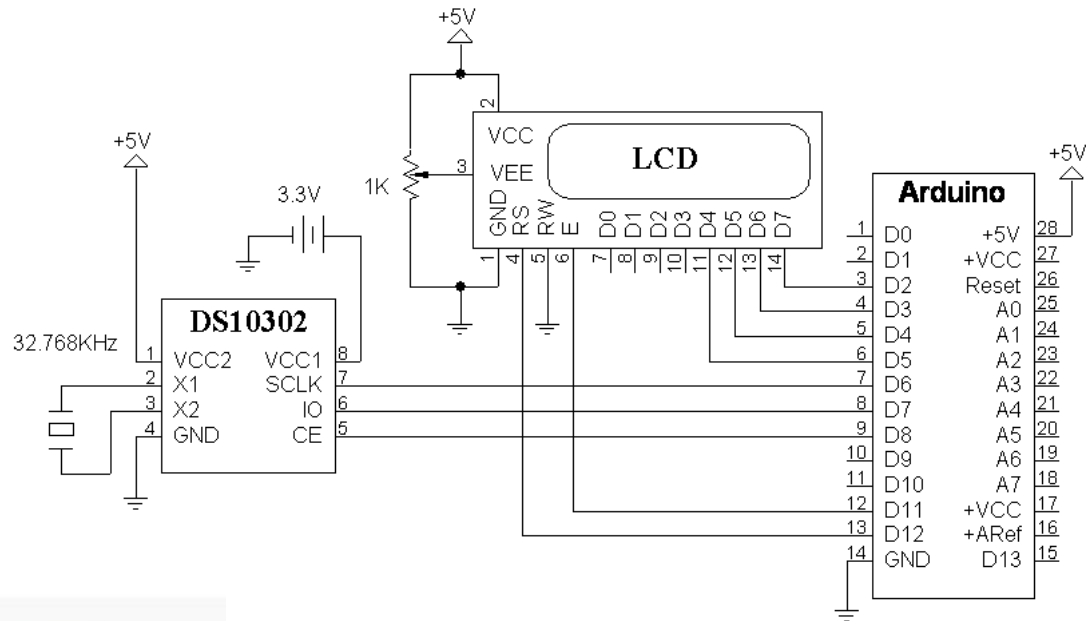
DS1302 8-Pin DIP (300-mil)

# DS1302 BLOCK DIAGRAM



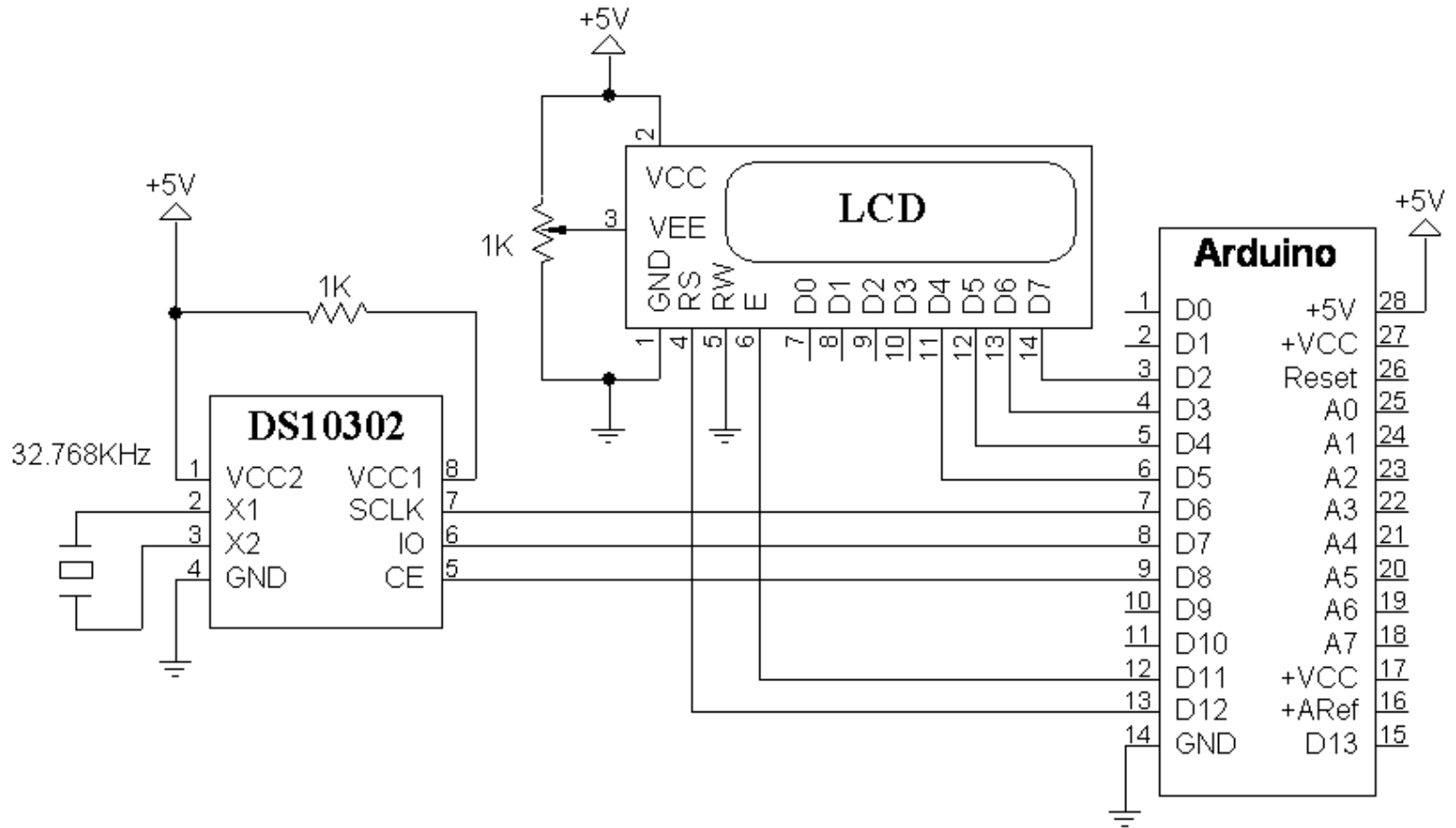
# การเชื่อมต่อ Arduino กับ Real-time clock DS1302

## และ LCD



# การเชื่อมต่อ Arduino กับ Real-time clock DS1302

## และ LCD



# ตำแหน่งข้อมูลรีจิสเตอร์ต่างๆเมื่อจะอ่านและเขียน

## RTC

| READ | WRITE | BIT 7   | BIT 6      | BIT 5       | BIT 4       | BIT 3   | BIT 2 | BIT 1 | BIT 0 | RANGE     |
|------|-------|---------|------------|-------------|-------------|---------|-------|-------|-------|-----------|
| 81h  | 80h   | CH      | 10 Seconds |             |             | Seconds |       |       |       | 00–59     |
| 83h  | 82h   |         | 10 Minutes |             |             | Minutes |       |       |       | 00–59     |
| 85h  | 84h   | 12/24   | 0          | 10<br>AM/PM | Hour        | Hour    |       |       |       | 1–12/0–23 |
| 87h  | 86h   | 0       | 0          | 10 Date     |             | Date    |       |       |       | 1–31      |
| 89h  | 88h   | 0       | 0          | 0           | 10<br>Month | Month   |       |       |       | 1–12      |
| 8Bh  | 8Ah   | 0       | 0          | 0           | 0           | 0       | Day   |       |       | 1–7       |
| 8Dh  | 8Ch   | 10 Year |            |             |             | Year    |       |       |       | 00–99     |
| 8Fh  | 8Eh   | WP      | 0          | 0           | 0           | 0       | 0     | 0     | 0     | —         |
| 91h  | 90h   | TCS     | TCS        | TCS         | TCS         | DS      | DS    | RS    | RS    | —         |

# ตำแหน่งข้อมูลรีจิสเตอร์ต่างๆเมื่อจะอ่านและเขียน

## CLOCK BURST

|     |     |
|-----|-----|
| BFh | BEh |
|-----|-----|

## RAM

|     |     |  |        |
|-----|-----|--|--------|
| C1h | C0h |  | 00-FFh |
| C3h | C2h |  | 00-FFh |
| C5h | C4h |  | 00-FFh |
| .   | .   |  | .      |
| .   | .   |  | .      |
| .   | .   |  | .      |
| FDh | FCh |  | 00-FFh |

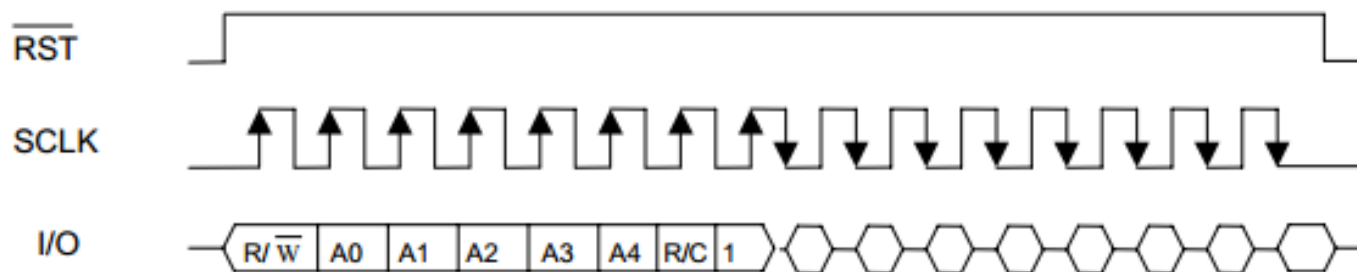
## RAM BURST

|     |     |
|-----|-----|
| FFh | FEh |
|-----|-----|

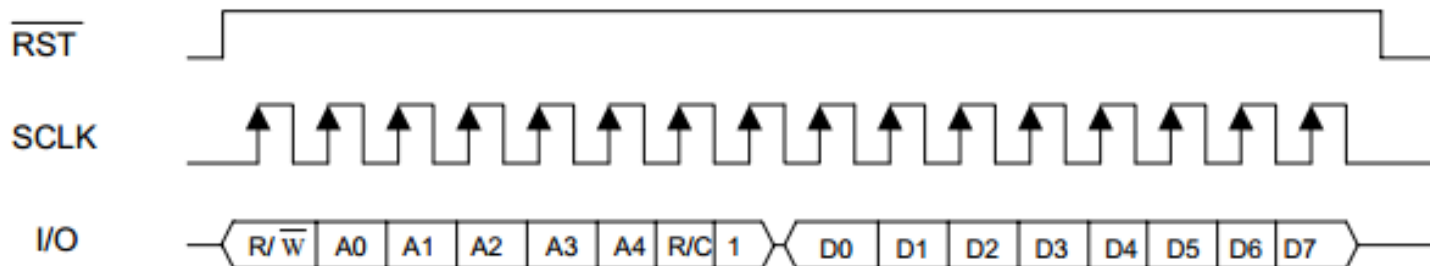


# การอ่านและเขียนข้อมูล

## SINGLE BYTE READ



## SINGLE BYTE WRITE



In burst mode,  $\overline{\text{RST}}$  is kept high and additional SCLK cycles are sent until the end of the burst.

# REGISTER ADDRESS/DEFINITION

## REGISTER ADDRESS

### A. CLOCK

|     |   |   |   |   |   |   |   |         |
|-----|---|---|---|---|---|---|---|---------|
|     | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0       |
| SEC | 1 | 0 | 0 | 0 | 0 | 0 | 0 | RD<br>W |

|     |   |   |   |   |   |   |   |         |
|-----|---|---|---|---|---|---|---|---------|
| MIN | 1 | 0 | 0 | 0 | 0 | 0 | 1 | RD<br>W |
|-----|---|---|---|---|---|---|---|---------|

|    |   |   |   |   |   |   |   |         |
|----|---|---|---|---|---|---|---|---------|
| HR | 1 | 0 | 0 | 0 | 0 | 1 | 0 | RD<br>W |
|----|---|---|---|---|---|---|---|---------|

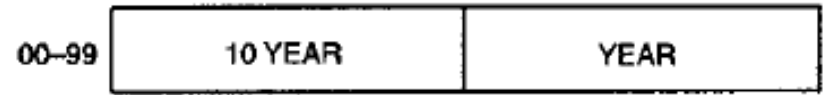
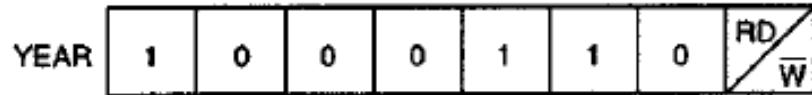
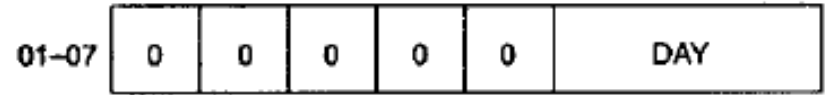
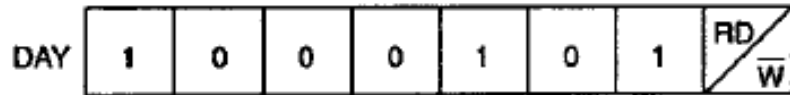
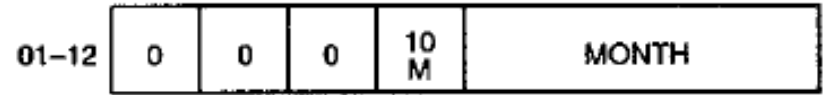
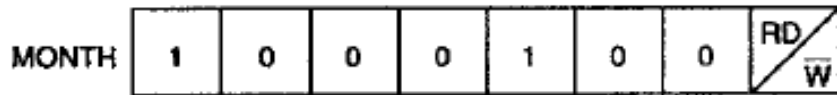
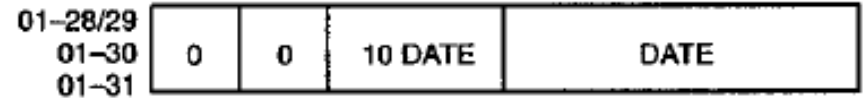
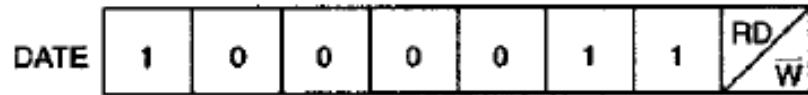
## REGISTER DEFINITION

|       |    |        |     |
|-------|----|--------|-----|
| 00-59 | CH | 10 SEC | SEC |
|-------|----|--------|-----|

|       |   |        |     |
|-------|---|--------|-----|
| 00-59 | 0 | 10 MIN | MIN |
|-------|---|--------|-----|

|       |     |   |     |    |    |
|-------|-----|---|-----|----|----|
| 01-12 | 12/ | 0 | 10  | HR | HR |
| 00-23 | 24  |   | A/P |    |    |

# REGISTER ADDRESS/DEFINITION



# ไลบรารี DS1302.h

Download มาจาก

- [http://blog.trendsmix.com/?attachment\\_id=148](http://blog.trendsmix.com/?attachment_id=148)

## การใช้งาน

1. กำหนดขาสัญญาณที่จะต่อกับ Arduino

```
uint8_t CE_PIN = 5;
```

```
uint8_t IO_PIN = 6;
```

```
uint8_t SCLK_PIN = 7;
```

## ไลบรารี DS1302.h (ต่อ)

2. สร้างออบเจ็ค DS1302

```
DS1302 rtc(CE_PIN, IO_PIN, SCLK_PIN);
```

3. Initialize a new chip by turning off write protection and clearing the clock halt flag. These methods needn't always be called. See the DS1302 datasheet for details.

```
rtc.write_protect(false);
```

```
rtc.halt(false);
```

4. Make a new time object to set the date and time

```
Time t(2009, 5, 19, 21, 16, 37, 3);
```

## ไลบรารี DS1302.h (ต่อ)

5. สร้างออบเจ็ค DS1302

```
DS1302 rtc(CE_PIN, IO_PIN, SCLK_PIN);
```

6. การตั้งวันเวลา Monday, Jan 13, 2014 at 13:30:00.

```
Time t(2014, 1, 13, 13, 30, 00, 2);
```

```
rtc.time(t);
```

7. การอ่านค่าเวลาจาก DS1302

```
Time t = rtc.time();
```

# โปรแกรม

```
#include <stdio.h>
#include <string.h>
#include <LiquidCrystal.h>
#include <DS1302.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

/* Set the appropriate digital I/O pin connections */
uint8_t CE_PIN   = 8;
uint8_t IO_PIN   = 7;
uint8_t SCLK_PIN = 6;

/* Create buffers */
char buf[50];
char day[10];

/* Create a DS1302 object */
DS1302 rtc(CE_PIN, IO_PIN, SCLK_PIN);
```

# โปรแกรม (ต่อ)

```
void print_time()
{
    /* Get the current time and date from the chip */
    Time t = rtc.time();

    /* Name the day of the week */
    memset(day, 0, sizeof(day)); /* clear day buffer */
    switch (t.day) {
        case 1: strcpy(day, "Sunday");      break;
        case 2: strcpy(day, "Monday");     break;
        case 3: strcpy(day, "Tuesday");    break;
        case 4: strcpy(day, "Wednesday");  break;
        case 5: strcpy(day, "Thursday");   break;
        case 6: strcpy(day, "Friday");     break;
        case 7: strcpy(day, "Saturday");   break;
    }
}
```



## โปรแกรม (ต่อ)

```
/* Format the time and date and insert into the temporary buffer */
snprintf(buf, sizeof(buf), "%s %04d-%02d-%02d
    %02d:%02d:%02d",
    day,
    t.yr, t.mon, t.date,
    t.hr, t.min, t.sec);

/* Print the formatted string to serial so we can see the time */
Serial.println(buf);
}
```

# โปรแกรม (ต่อ)

```
void setup()
{
  Serial.begin(9600);

  lcd.begin(16, 2); // set up the LCD's number of columns and rows:

  lcd.print("Real time clock"); // Print a message to the LCD.

  /* Initialize a new chip by turning off write protection and clearing the clock halt flag. These methods needn't always
     be called. See the DS1302 datasheet for details. */
  rtc.write_protect(false);
  rtc.halt(false);

  /* Make a new time object to set the date and time */
  /*  Monday, Jan 13, 2014 at 13:30:37.          */
  Time t(2014, 1, 13, 13, 30, 37, 2);

  /* Set the time and date on the chip */
  rtc.time(t);
  delay(1000);
}
```

## โปรแกรม (ต่อ)

```
/* Loop and print the time every second */  
void loop()  
{  
  print_time();  
  delay(1000);  
}
```

# โปรแกรม (ต่อ)