

Philips LPC2138 Weather Station Abstract

We are all influenced by the weather. Most of us pay particular attention to the weather man on television. Most of us have an outdoor thermometer. Many of us even considered buying their own personal weather station. I happen to be one of those people.

I have selected three measurements for my weather station. These include outside temperature, wind direction and wind speed.

One of the major objectives of my design was to minimize the number of electronic parts and to simplify the design using easily obtainable parts.

Photo 1 is a photograph of the finished project.

Photo 2 illustrates the end unit construction, testing, and software development.

The display module is a Futaba NA202SD01 Vacuum Fluorescent Display (VFD), which only required a single 5 volts DC supply.

I used the Philips 2138 microcontroller and the Kiel MSD2130 development board to quickly prototype my weather station.

Schematic 1 is a modified schematic of the Kiel development board to include my circuits to support my weather station.

The software developed includes interrupt routines, utilities, and the application software itself.



Photo 1 - The LPC2138 Weather Station



Photo 2 – The Weather Station Construction

Listing 1 is the main body of the application program illustrating the initialization and recurring operational program (i.e., the infinite loop).

Listing 1 - Main (See complete listings for interrupt and utilities)

```
int main (void) {
    float tmp1, tmp2, tmp3;
    float ad0, ad1;
    unsigned int n;
    unsigned char task_active = 0x80;
    // lets try some tasking
    // b x x | x x | x x | x x
    //  M   | T1  | T2  | RTC   1st indicates "active"
    init_registers();
    init_serial();           // Initialize Serial Interface
    init_timer();           // init timer 0
    init_timer_1();         // init timer 1
    init_clock();           // init real-time clock
    printf("\n");           // clear displar
    while (ON) {           // while microprocess is on, Loop forever
        if ( ((task_active)&(0x80)) == 0x80 ) { // test for active
            set_LED(ZERO,ON); // indicate active
            task_active &= 0x7F; // clear active bit (preserve all others)
        }
        else {
            set_LED(ZERO,OFF); // indicate inactive
            task_active |= 0x80; // set active bit
        }
        disp_lum(1);         // display lighting (0=off, 1=20%...,5=100%
        disp_set_cursor(0,0);
        wind_dir_char();
        speed=wind_speed(timer_0_capture);
        printf(" Wind %s",wind_dir_str);
        printf(" %2.1f MPH ",speed);
        disp_set_cursor(1,0);
        printf("Temp %3.0f",air_temp_F(air_temp(tval)));
        putchar(0xDF); printf("F");
        printf(" "); wr_byte(2,HOUR);
        printf(":"); wr_byte(2,MIN);
        printf(":"); wr_byte(2,SEC);

        last_sec = SEC;
        while (last_sec == SEC);
    }
}
```