

SISTEMA MINIMO CON ATMEGA8 / ATMEGA168

Las pantallas alfanuméricas LCD de 16 caracteres y dos filas con interface paralela, de uso común en el mercado, tienen el inconveniente de ser muy sensibles a los ruidos electromagnéticos, lo que las inhabilita para ser empleadas en la mayoría de los equipos donde se manejen niveles considerables de potencia, mediante algún tipo de convertidor electrónico.

Los ruidos e interferencias llegan a estas pantallas, a través de los hilos de datos, que en la mayoría de los casos tienen más de 40 mm de longitud, creando problemas aun estando debidamente apantallados.

Para erradicar este problema, hemos diseñado un display con comunicación serie Rs485 / UART. El bus rs485 tiene una gran inmunidad contra los ruidos, por ser en principio un bus de corriente.



Lo que hemos hecho es adosar una pantalla standard de fácil adquisición en el mercado, a una placa con un micro atmega8, cuya Usart se ha conectado a un driver de bus Max485.

En la placa, como muestra el circuito, se han habilitado dos entradas analógicas y dos salidas digitales, lo que permite que ella pueda funcionar como una simple pantalla de visualización, o como un pequeño sistema de control operado de forma local o remota. Por ejemplo, se puede convertir en una incubadora solo añadiendo un relé de estado sólido, y una ntc o LM35 como sensor de temperatura y un potenciómetro para la consigna de temperatura.

Esquemas eléctricos.

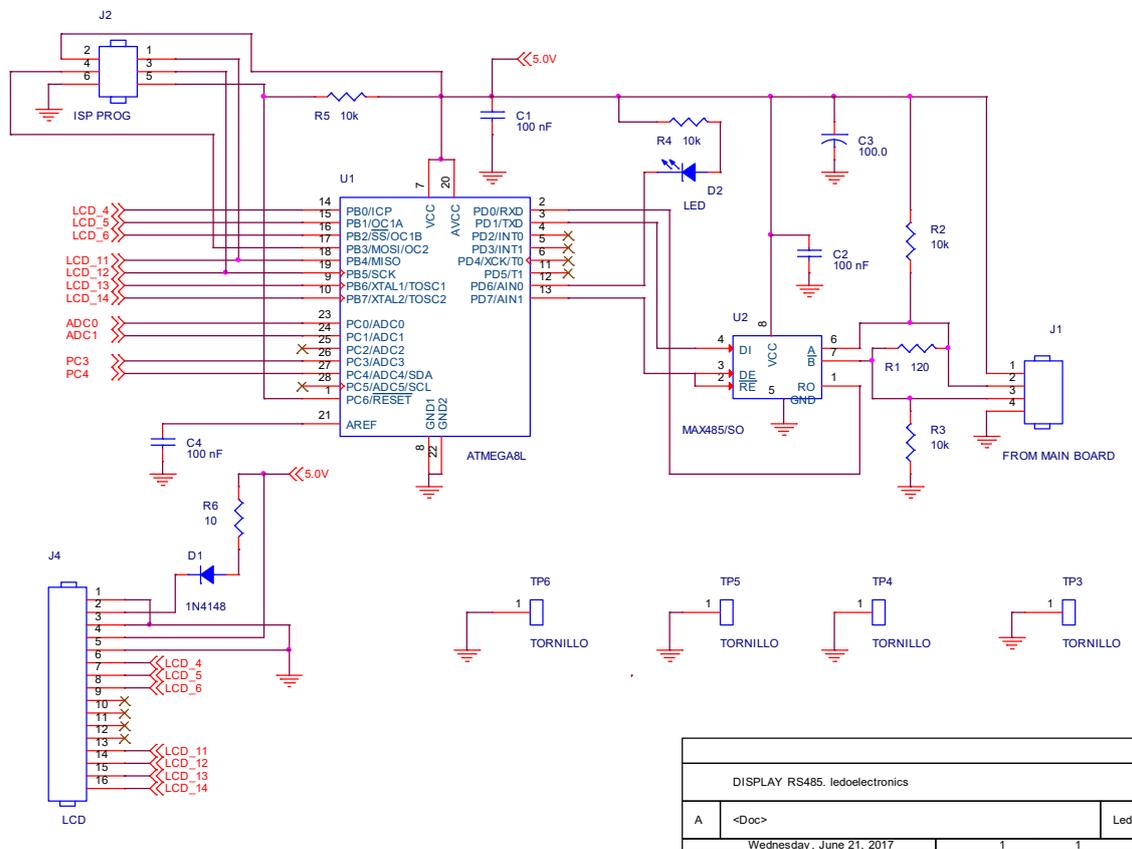


Fig1. Sistema mínimo con atmega8.

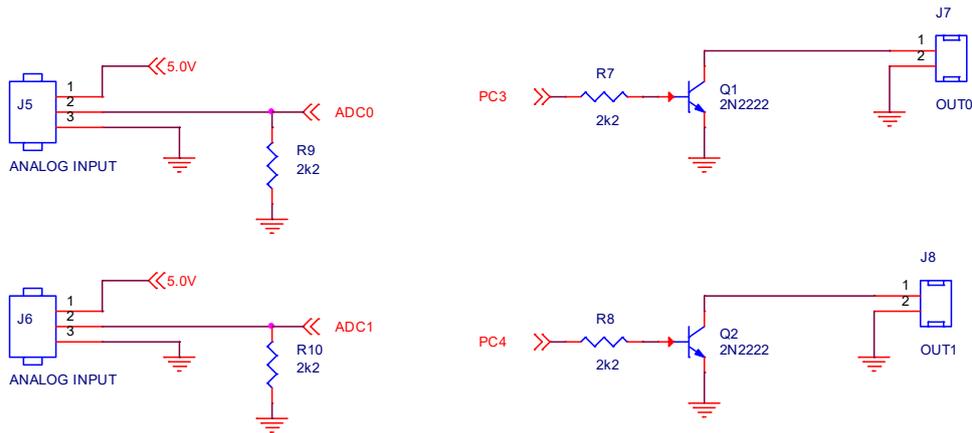


Fig2. Entradas salidas opcionales.

La placa se sirve por defecto con el firmware de control de temperatura para una incubadora con LM35 como detector de temperatura, y en la pantalla se muestra El valor de la consigna y de la temperatura actual.

Se puede servir bajo pedido con el firmware correspondiente a un display Rs485. En cualquier caso, el atmega8 puede ser programado in system mediante el conector ISP J2.

Comunicación con la pantalla Rs485

Los parámetros de la interface serie, para comunicarse con la pantalla deben ser los siguientes:

- Velocidad de transferencia: 38400 bit/seg.
- Formato de la trama: 8, 1, sin paridad.
- Dirección Rs485: 23

A continuación, se muestra un ejemplo de trabajo con la pantalla desde una placa basada en un microcontrolador xmega32.

Primeramente definimos las variables y la función de transmisión de datos:

```
#define TX_BUFFER_SIZE 36 //longitud del buffer de transmisión
uint8_t lcd_buffer_H[16]; //contenido fila alta del display
uint8_t lcd_buffer_L[16]; //contenido fila baja del display
uint8_t tx_buffer[TX_BUFFER_SIZE]; //buffer de transmisión
void send_to_lcd_rs485(uint8_t);
```

```

/* Transmite comandos y datos al lcd rs485 *****
** cmd = 1 Inicialización del display *****
** cmd = 2 Envío de datos a visualizar *****
** Antes de usar la función, es necesario tener los datos listos en los buffers ***
** lcd_buffer_L y lcd_buffer_H ***
*****/
void send_to_lcd_rs485(uint8_t cmd)
{
    int8_t i;
    int16_t cksum = 0;
    uint8_t * s;
    uint8_t * d;
    tx_buffer[0] = 23; //dirección Rs485 del display
    tx_buffer[2] = cmd;
    s = &lcd_buffer_H[0];
    d = &usart.tx_buffer[2];
    for(int8_t k=0;k<16;k++)
        * d++ = * s++;
    s = &lcd_buffer_L[0];
    for(int k=0;k<16;k++)
        * d++ = * s++;
    PORTE.OUTSET = DE; //rs485 bus transmit enable
    _delay_us(10); //espera estabilización del Max485
    for(i=0;i<TX_BUFFER_SIZE -2;i++)
        cksum += tx_buffer[i];
    tx_buffer[TX_BUFFER_SIZE -2] = (uint8_t) (cksum % 256);
    tx_buffer[TX_BUFFER_SIZE -1] = (uint8_t) (cksum >> 8);
    for (i=0;i<TX_BUFFER_SIZE;i++)
    {
        // Wait for empty transmit buffer /
        while ( !(USARTE0.STATUS & (1<<USART_DREIF_bp )))
            ;
        // Put data into buffer, sends the data /
        USARTE0.DATA = tx_buffer[i];
    }
    //espera a que se transmitan todos los bits
    while ( !( USARTE0.STATUS & (1<<USART_TXCIF_bp ) ))
        ;
    USARTE0.STATUS |= (1<<USART_TXCIF_bp ); //flag clear

    PORTE.OUTCLR = DE; // pone modo escucha, en red rs485
} // end rs485_tx()

```

Podemos usarla como sigue, desde cualquier parte ejecutable del programa

```

strcpy(lcd_buffer_H, "ledoelectronics ");
sprintf(lcd_buffer_L, " Hola a todos ");
tx_buffer[0] = 23;
send_to_lcd_rs485(2);

```

