

ATMEGA Mini Project Board

Contents

OVERVIEW	3
ATMEGA Mini Project Board	3
FEATURES	3
CIRCUIT DIAGRAM	4
PIN CONFIGURATION	5
CODES	6
1. LED BLINKING	6
2. LCD	6
RELATED PRODUCT	8

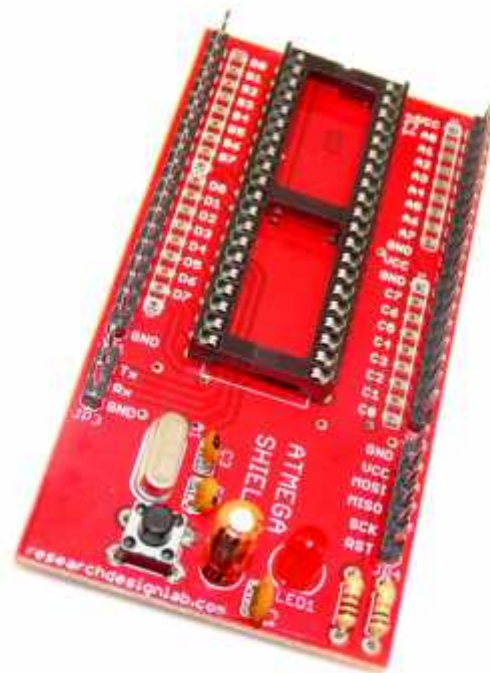
OVERVIEW

ATMEGA Mini Project Board

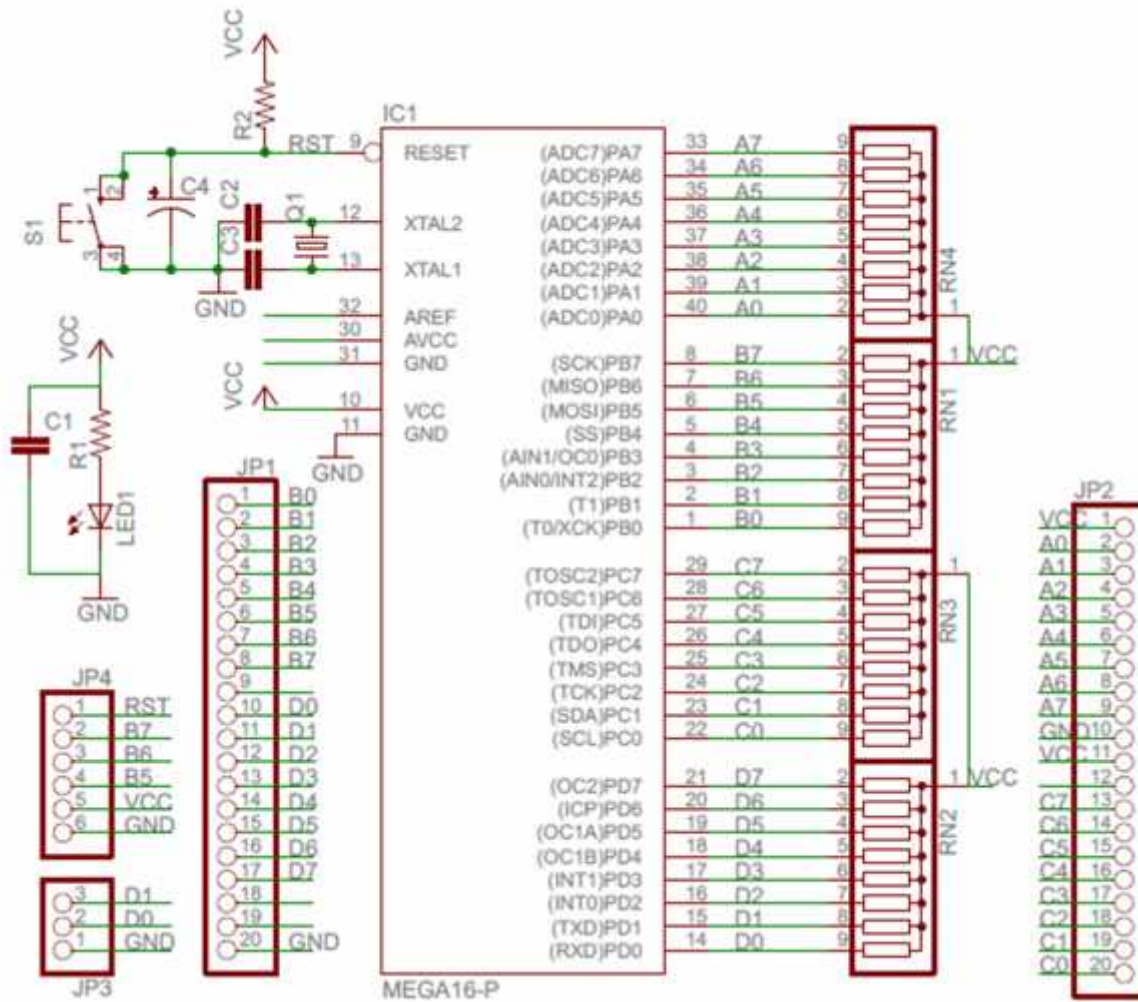
If you are learning microcontrollers or want to quickly develop embedded solution based on standard AVR core, this board will help you quick start with the application by giving you access to everything required to run the Atmega16 microcontroller. With this board you can develop and prototype with any of Microchip's 40 pin AVR microcontrollers. . The boards have User button and status LED.

FEATURES

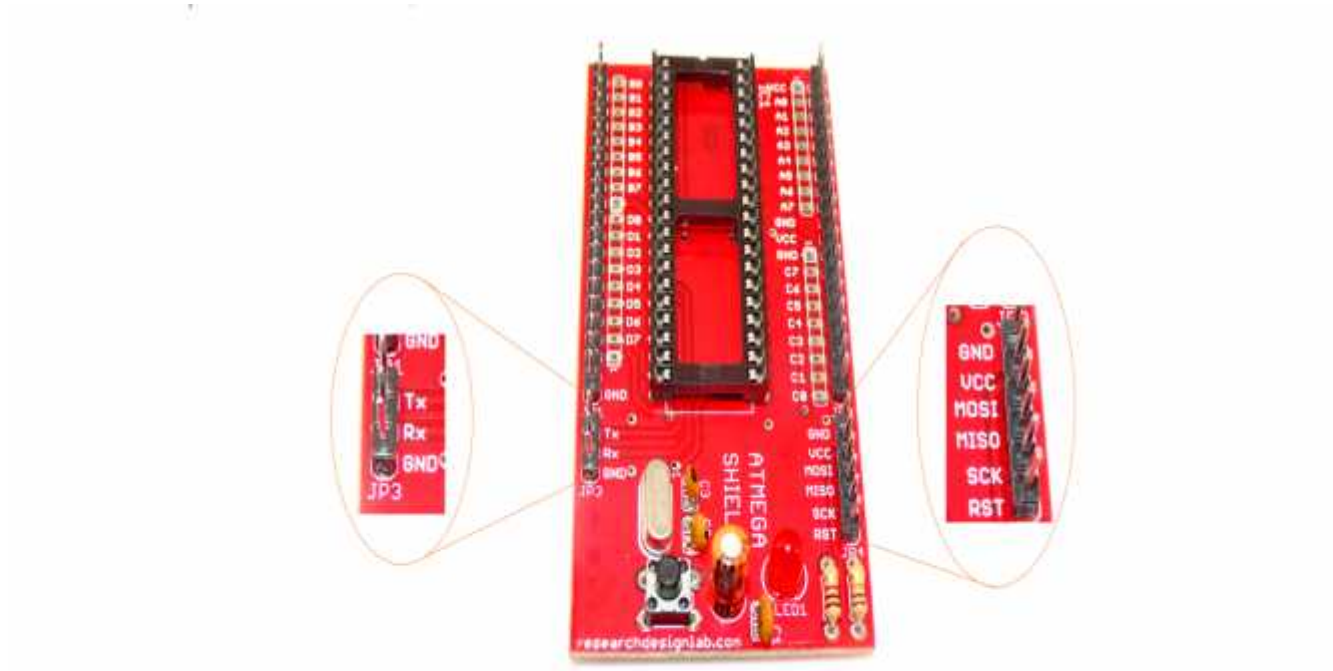
- Quartz crystal of 16 MHz
- Reset button
- Power indicating LED
- External TX,RX,GND
- On-board ISP Programmer



CIRCUIT DIAGRAM



PIN CONFIGURATION



1. **TX**, **RX** and **GND** is used to provide serial communication UART
2. **MOSI** - Master Out Slave In; **MISO** - Master In Slave Out; **SCK** - Clock signal from master to slave; **SS** - Slave Select signal selects slave devices. SPI is a bus and can have multiple devices on the bus. Because of a clock signal present, SPI can be operated faster than UART.

CODES

1. LED BLINKING

```
#define F_CPU 1000000UL // using internal 1MHZ frequency
#include <avr/io.h> //initialize input output
#include <util/delay.h> //initialize delay function

int main(void)
{
    DDRB = 0xFF; //Makes PORTB as Output
    while(1) //infinite loop
    {
        PORTB = 0xFF; //Turns ON All LEDs
        _delay_ms(1); //1 second delay
        PORTB= 0x00; //Turns OFF All LEDs
        _delay_ms(1); //1 second delay
    }
}
```

2. LCD

```
# define F_CPU 1000000UL
#include <avr/io.h>
#include <util/delay.h>
#include <string.h> //initialize the string function

#define LCD_PORT PORTB
#define RS PC0 //initialize register select as PC0 pin
#define EN PC1 //initialize enable pin as PC1

void CMD_WRT(unsigned char val)
{
    PORTB=val;
    PORTC = PORTC & ~(1<<RS);
    _delay_ms(1); // here we provide a delay of 1 sec
    PORTC = PORTC | ((1<<EN)); //make enable pin high
    _delay_ms(1);
    PORTC = PORTC & ~(1<<EN); //make enable pin low
}

void DATA_WRT(unsigned char ch)
{
    PORTB = ch; //initializing character "ch" to port B
    PORTC = PORTC | ((1<<RS)); //make register select pin high
    _delay_ms(1);
}
```

```
        PORTC = PORTC | ((1<<EN)); //make enable pin high
        _delay_ms(1);
        PORTC = PORTC & ~(1<<EN); //make enable pin low
    }
void LCD_WRT( char *string)
{
    while(*string)
        DATA_WRT(*string++);    //will write the strings
}

int main(void)
{
    //setting the display of the LCD unsigned char
    CMD[]= {0x38,0x01,0x0f,0x06,0x80},TEMP1,i;

    DDRB=0xFF;                //make PORTB as output
    DDRC = 0xFF; //(1<<RS)|(1<<EN); //make PORTC as output
    _delay_ms(10);            //provide the delay of 10ms

    for(i=0;i<5;i++)
    {
        TEMP1=CMD[i]; //it will place the command in cmd array
        CMD_WRT(TEMP1); //it will write all the cmd that is in
                        //the cmd array
    }

    while(1)
    {

        CMD_WRT(0X01); //clear display
        CMD_WRT(0X80); // blink the cursor in 1st row
        LCD_WRT("  --RDL--"); //display RDL in lcd
        CMD_WRT(0XC0); //to use 2nd row of lcd
        LCD_WRT(" LCD_DISPLAY"); //display LCD_DISPLAY in
                                //lcd
        _delay_ms(1000); //delay of 1sec

    }

    return 0;
}
```

RELATED PRODUCT

Atmega16/32/64 Project Board



ATMEGA 16/32/64 Development Board-USB



ATmega Programmer-USB



ISP Atmel Programmer

