

# **LAMPIRAN**

## Pengaturan awal program

The image displays four screenshots of the AVR Studio 4 software interface, illustrating the initial setup and configuration of a AVR microcontroller project.

**Screenshot 1 (Top Left):** Shows the main project configuration window. It includes tabs for USART, Analog Comparator, ADC, SPI, I2C, 1-Wire, and TWI (I2C). Under the "Project Information" tab, "Bit-Banged" is selected. Below the tabs, there are buttons for Alphanumeric LCD and Graphic LCD. The "Chip" tab is selected, showing "ATmega8" as the chip type and a clock frequency of "16,000000 MHz". There are also dropdowns for "Check Reset Source" and "Program Type" set to "Application".

**Screenshot 2 (Top Right):** Shows the configuration for an LCD module. It includes tabs for USART, Analog Comparator, ADC, SPI, I2C, 1-Wire, and TWI (I2C). Under the "Project Information" tab, "Bit-Banged" is selected. Below the tabs, there are buttons for Alphanumeric LCD and Graphic LCD. A checked checkbox "Enable Alphanumeric LCD Support" is present. The "Controller Type" is set to "HD44780" and "Characters/Line" is set to "16". The "Connections" section shows the pin mapping for the LCD module:

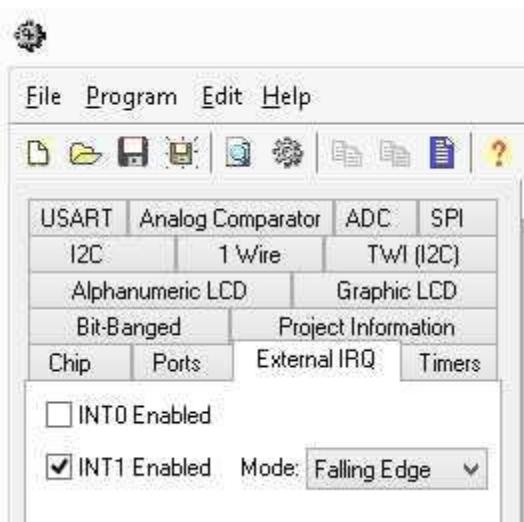
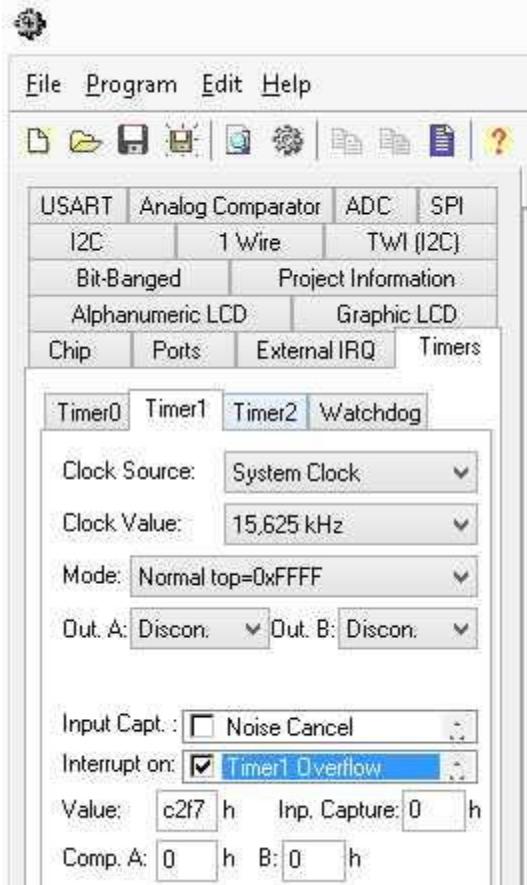
| Pin | Port  | Bit |
|-----|-------|-----|
| RS  | PORTE | 0   |
| RD  | PORTE | 1   |
| EN  | PORTE | 2   |
| D4  | PORTE | 4   |
| D5  | PORTE | 5   |
| D6  | PORTE | 6   |
| D7  | PORTE | 7   |

**Screenshot 3 (Bottom Left):** Shows the port configuration for Port B, Port C, and Port D. It includes tabs for USART, Analog Comparator, ADC, SPI, I2C, 1-Wire, and TWI (I2C). Under the "Project Information" tab, "Bit-Banged" is selected. Below the tabs, there are buttons for Alphanumeric LCD and Graphic LCD. The "Ports" tab is selected, showing the following port settings:

| Port   | Data Direction | Pullup/Output Value |
|--------|----------------|---------------------|
| Port B | Bit 0 In       | T Bit 0             |
| Port B | Bit 1 In       | T Bit 1             |
| Port B | Bit 2 In       | T Bit 2             |
| Port B | Bit 3 In       | P Bit 3             |
| Port B | Bit 4 In       | P Bit 4             |
| Port B | Bit 5 In       | P Bit 5             |
| Port B | Bit 6 In       | T Bit 6             |

**Screenshot 4 (Bottom Right):** Shows the port configuration for Port B, Port C, and Port D. It includes tabs for USART, Analog Comparator, ADC, SPI, I2C, 1-Wire, and TWI (I2C). Under the "Project Information" tab, "Bit-Banged" is selected. Below the tabs, there are buttons for Alphanumeric LCD and Graphic LCD. The "Ports" tab is selected, showing the following port settings:

| Port   | Data Direction | Pullup/Output Value |
|--------|----------------|---------------------|
| Port B | Bit 0 In       | T Bit 0             |
| Port B | Bit 1 In       | T Bit 1             |
| Port B | Bit 2 In       | T Bit 2             |
| Port B | Bit 3 In       | P Bit 3             |
| Port B | Bit 4 In       | T Bit 4             |
| Port B | Bit 5 In       | T Bit 5             |
| Port B | Bit 6 In       | T Bit 6             |
| Port B | Bit 7 In       | T Bit 7             |



## **Listing program**

```
*****  
Chip type      : ATmega8  
Program type   : Application  
AVR Core Clock frequency: 16,000000 MHz  
Memory model   : Small  
External RAM size : 0  
Data Stack size : 256  
*****  
  
#include <mega8.h>  
  
// Alphanumeric LCD functions  
#include <alcd.h>  
// library delay  
#include <delay.h>  
// library standart i/o  
#include <stdio.h>  
// definisi tombol  
#define s1 PINC.3  
#define s2 PINC.4  
#define s3 PINC.5  
// definisi sensor  
#define ir PIND.3  
// display lcd buffer  
char buff[33];  
// global variable  
int count=0;    // pencacah input sensor  
int rpm=0,time=0; // nilai rpm dan waktu
```

```

bit mode_start=0; // mode start/stop pembacaan rpm
int i=0;          // index array
int timer=90;     // batas waktu

eeprom int data_log[6]={0,0,0,0,0,0}; // eeprom penyimpan data rpm

/*
rumus timer:
(16bit+1)-(1detik*(xtal/prescaller)
TCNT: (65535+1)+(1*(16mhz/1024))
TCNT: 49911
jadikan HEXADESIMAL
TCNT: C2F7
*/
// Timer1 overflow interrupt service routine
interrupt [TIM1_OVF] void timer1_ovf_isr(void)
{
// Reinitialize Timer1 value
TCNT1H=0xC2F7 >> 8;
TCNT1L=0xC2F7 & 0xff;
// Place your code here
rpm=count*60;
count=0;
if(mode_start==1)time++;
}

// External Interrupt 1 service routine

```

```
interrupt [EXT_INT1] void ext_int1_isr(void)
{
    // Place your code here
    if(mode_start==1)count++;
}

// Declare your global variables here

// program melihat data loger
void lihat_log(){
    int index=0;
    delay_ms(200);
    lcd_clear();
    while(1){
        lcd_clear();
        lcd_gotoxy(0,0);
        lcd_putsf("SHOW DATA LOG");
        lcd_gotoxy(0,1);
        sprintf(buff,"Log%d : %d",index+1,data_log[index]);
        lcd_puts(buff);

        if(!s2)index++; // menambah index
        if(!s3)index--; // mengurangi index
        // pembatas nilai
        if(index>5)index=0;
        if(index<0)index=5;
        if(!s1)break;

        delay_ms(150); // delay refresh
}
```

```
}

delay_ms(200);

}

// program display rpm

void display_rpm(){

lcd_clear();

lcd_gotoxy(0,0);

sprintf(buff,"RPM: %d",rpm);

lcd_puts(buff);

lcd_gotoxy(0,1);

sprintf(buff,"S:%d T:%d I:%d",ir,time,i);

lcd_puts(buff);

// jika s1 ditekan mode start aktif

if(!s1) mode_start=1;

// jika nilai time sudah sama dengan timer maka mode stop

if(time>timer){

mode_start=0;

count=0;

time=0;

i=0;

}

// panggil lihat log

if(!s2&&mode_start==0)lihat_log();

// jika mode start aktif maka menyimpan data ke eeprom

if(mode_start==1){
```

```
// jika nilai rpm tidak sama dengan data memory maka data memory diisi oleh data
// rpm

if(rpm!=data_log[i])data_log[i]=rpm;

i=time/15; // simpan per 15 detik

}

delay_ms(50); // delay refresh
}

void main(void)
{
// Declare your local variables here

// Input/Output Ports initialization
// Port B initialization
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=T State2=T State1=T State0=T
PORTB=0x00;
DDRB=0x00;

// Port C initialization
// Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State6=T State5=P State4=P State3=P State2=T State1=T State0=T
PORTC=0x38;
DDRC=0x00;

// Port D initialization
```

```
// Func7=In Func6=In Func5=In Func4=In Func3=In Func2=In Func1=In Func0=In
// State7=T State6=T State5=T State4=T State3=P State2=T State1=T State0=T
PORTD=0x08;
DDRD=0x00;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
TCCR0=0x00;
TCNT0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 15,625 kHz
// Mode: Normal top=0xFFFF
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge
// Timer1 Overflow Interrupt: On
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x05;
TCNT1H=0xC2;
TCNT1L=0xF7;
ICR1H=0x00;
ICR1L=0x00;
```

```
OCR1AH=0x00;  
OCR1AL=0x00;  
OCR1BH=0x00;  
OCR1BL=0x00;  
  
// Timer/Counter 2 initialization  
// Clock source: System Clock  
// Clock value: Timer2 Stopped  
// Mode: Normal top=0xFF  
// OC2 output: Disconnected  
ASSR=0x00;  
TCCR2=0x00;  
TCNT2=0x00;  
OCR2=0x00;  
  
// External Interrupt(s) initialization  
// INT0: Off  
// INT1: On  
// INT1 Mode: Falling Edge  
GICR|=0x80;  
MCUCR=0x08;  
GIFR=0x80;  
  
// Timer(s)/Counter(s) Interrupt(s) initialization  
TIMSK=0x04;  
  
// USART initialization  
// USART disabled  
UCSRB=0x00;
```

```
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;

// ADC initialization
// ADC disabled
ADCSRA=0x00;

// SPI initialization
// SPI disabled
SPCR=0x00;

// TWI initialization
// TWI disabled
TWCR=0x00;

// Alphanumeric LCD initialization
// Connections are specified in the
// Project|Configure|C Compiler|Libraries|Alphanumeric LCD menu:
// RS - PORTD Bit 0
// RD - PORTD Bit 1
// EN - PORTD Bit 2
// D4 - PORTD Bit 4
// D5 - PORTD Bit 5
// D6 - PORTD Bit 6
// D7 - PORTD Bit 7
```

```
// Characters/line: 16

lcd_init(16);

// Global enable interrupts
#asm("sei")

lcd_clear();
lcd_gotoxy(0,0);
lcd_putsf("Bismillah");
lcd_gotoxy(0,1);
lcd_putsf("Tachometer");
delay_ms(1000);

while (1)
{
    // Place your code here
    display_rpm();
}

}
```

## Proses pengambilan data







