

LAMPIRAN A
LISTING PROGRAM

```
/******
```

```
This program was produced by the  
CodeWizardAVR V1.25.3 Professional  
Automatic Program Generator  
© Copyright 1998-2007 Pavel Haiduc, HP InfoTech s.r.l.  
http://www.hpinfotech.com
```

```
Project :  
Version :  
Date   : 11/21/2008  
Author : Lab Instrumentasi  
Company : UKM  
Comments:
```

```
Chip type      : ATmega16  
Program type   : Application  
Clock frequency : 11.059200 MHz  
Memory model   : Small  
External SRAM size : 0  
Data Stack size : 256
```

```
*****/
```

```
#include <mega16.h>  
#include <delay.h>  
#include <stdio.h>
```

```
#define kopi 0x00  
#define gula 0x01  
#define beras 0x02
```

```
#define kopi_buka PORTD=0x01; delay_ms(500); PORTD=0  
#define kopi_tutup PORTD=0x02; delay_ms(500); PORTD=0  
#define gula_buka PORTD=0x04; delay_ms(500); PORTD=0  
#define gula_tutup PORTD=0x08; delay_ms(500); PORTD=0  
#define beras_buka PORTD=0x10; delay_ms(500); PORTD=0  
#define beras_tutup PORTD=0x20; delay_ms(500); PORTD=0  
#define led_nyala PORTD=0x40;  
#define led_mati PORTD=0;
```

```
EEPROM int harga_kopi=1,harga_gula=2,harga_beras=3;  
int tempi,jumlah,hargatotal;  
char tempc,text[16],pilihan,cek[3];  
int gr;
```

```

// Alphanumeric LCD Module functions
#asm
.equ __lcd_port=0x15 ;PORTC
#endasm
#include <lcd.h>

#define ADC_VREF_TYPE 0x40

// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)!=0);
ADCSRA|=0x10;
return ADCW;
}

// Declare your global variables here

char keypad(void)
{
char scankey;
char keypressed =0;
DDRB = 0x0F;
PORTB = 0xFE;
scankey = PINB&0xf0;
switch (scankey)
{
case 0xE0 : keypressed = 1;
break;
case 0xD0 : keypressed = 2;
break;
case 0xB0 : keypressed = 3;
break;
case 0x70 : keypressed = 'A';
break;
}
key1:
scankey = PINB&0xf0;
if(scankey!=0xf0)
goto key1;

PORTB = 0xFD;

```

```

    delay_us(1);
scankey = PINB&0xf0;
switch (scankey)
{
    case 0xE0 : keypressed = 4;
    break;
    case 0xD0 : keypressed = 5;
    break;
    case 0xB0 : keypressed = 6;
    break;
    case 0x70 : keypressed = 'B';
    break;
}

```

```

key2:
scankey = PINB&0xf0;
if(scankey!=0xf0)
goto key2;

```

```

PORTB = 0xFB;
delay_us(1);
scankey = PINB&0xf0;
switch (scankey)
{
    case 0xE0 : keypressed = 7;
    break;
    case 0xD0 : keypressed = 8;
    break;
    case 0xB0 : keypressed = 9;
    break;
    case 0x70 : keypressed = 'C';
    break;
}

```

```

key3:
scankey = PINB&0xf0;
if(scankey!=0xf0)
goto key3;

```

```

PORTB = 0xF7;
delay_us(1);
scankey = PINB&0xf0;
switch (scankey)
{
    case 0xE0 : keypressed = '*';
    break;
}

```

```

    case 0xD0 : keypressed = '0';
    break;
    case 0xB0 : keypressed = '#';
    break;
    case 0x70 : keypressed = 'D';
    break;
    }

    key4:
    scankey = PINB&0xf0;
    if(scankey!=0xf0)
    goto key4;
    PORTB=0;
    return keypressed;

}

int berat (int input)
{
int output;

if(input<450)
output=0;
if(input>=450&&input<452)
output=5;
if(input==452)
output=10;
if(input>=453&&input<455)
output=15;
if(input==455)
output=20;
if(input>=456&&input<458)
output=25;
if(input>=458&&input<460)
output=30;
if(input==460)
output=35;
if(input>=461&&input<463)
output=40;
if(input==463)
output=45;
if(input==464)
output=50;
if(input==465)

```

```
output=55;
if(input>=466&&input<468)
output=60;
if(input>=468&&input<470)
output=65;
if(input==470)
output=70;
if(input>=471&&input<473)
output=75;
if(input>=473&&input<475)
output=80;
if(input==475)
output=85;
if(input>=476&&input<478)
output=90;
if(input==478)
output=95;
if(input==479)
output=100;
if(input==480)
output=105;
if(input==481)
output=110;
if(input>=482&&input<484)
output=115;
if(input==484)
output=120;
if(input==485)
output=125;
if(input==486)
output=130;
if(input==487)
output=135;
if(input>=488&&input<490)
output=140;
if(input>=490&&input<492)
output=145;
if(input>=492&&input<495)
output=150;
if(input==495)
output=155;
if(input>=496&&input<498)
output=160;
if(input>=498&&input<500)
output=165;
if(input>=500&&input<502)
```

```
output=170;
if(input==502)
output=175;
if(input>=503&&input<505)
output=180;
if(input>=505&&input<507)
output=185;
if(input==507)
output=190;
if(input>=508&&input<511)
output=195;
if(input==511)
output=200;
if(input>=512&&input<514)
output=205;
if(input>=514&&input<516)
output=210;
if(input>=516&&input<518)
output=215;
if(input==518)
output=220;
if(input>=519&&input<521)
output=225;
if(input==521)
output=230;
if(input==522)
output=235;
if(input==523)
output=240;
if(input==524)
output=245;
if(input>=525&&input<528)
output=250;
if(input>=528)
output=250;
```

```
return output;
//return input;
}
```

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

```

// Input/Output Ports initialization
// Port A 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
PORTA=0x00;
DDRA=0x00;

// 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
// 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
PORTC=0x00;
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=T State2=T State1=T State0=T
PORTD=0x00;
DDRD=0xff;

// Timer/Counter 0 initialization
// Clock source: System Clock
// Clock value: Timer 0 Stopped
// Mode: Normal top=FFh
// OC0 output: Disconnected
TCCR0=0x00;
TCNT0=0x00;
OCR0=0x00;

// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: Timer 1 Stopped
// Mode: Normal top=FFFFh
// OC1A output: Discon.
// OC1B output: Discon.
// Noise Canceler: Off
// Input Capture on Falling Edge

```



```

// Timer 1 Overflow Interrupt: Off
// Input Capture Interrupt: Off
// Compare A Match Interrupt: Off
// Compare B Match Interrupt: Off
TCCR1A=0x00;
TCCR1B=0x00;
TCNT1H=0x00;
TCNT1L=0x00;
ICR1H=0x00;
ICR1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;

// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: Timer 2 Stopped
// Mode: Normal top=FFh
// OC2 output: Disconnected
ASSR=0x00;
TCCR2=0x00;
TCNT2=0x00;
OCR2=0x00;

// External Interrupt(s) initialization
// INT0: Off
// INT1: Off
// INT2: Off
MCUCR=0x00;
MCUCSR=0x00;

// Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00;

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

// ADC initialization
// ADC Clock frequency: 691.200 kHz
// ADC Voltage Reference: AVCC pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;

```

```

ADCSRA=0x84;

// LCD module initialization
lcd_init(16);

kopi_tutup;
gula_tutup;
beras_tutup;

lcd_putsf("Masukkan Pilihan");
delay_ms(1000);
lcd_clear();
lcd_putsf("1=kopi 2=gula 3=beras");

while (1)
{
// Place your code here
//=====
=====
cek1:
tempc=keypad();
switch(tempc)
{
case 1:lcd_clear();lcd_putsf("Pilihan kopi");pilihan=kopi;goto cek2;break;
case 2:lcd_clear();lcd_putsf("Pilihan gula");pilihan=gula;goto cek2;break;
case 3:lcd_clear();lcd_putsf("Pilihan beras");pilihan=beras;goto cek2;break;
}
goto cek1;

//=====
=====

cek2:
delay_ms(1000);
lcd_clear();
lcd_putsf("Masukkan jumlah");
delay_ms(1000);
tempi=0;
//=====
=====

cek3:
sprintf(text,"%5u A-Reset",tempi);
lcd_gotoxy(0,1);
lcd_puts(text);
tempc=keypad();
switch(tempc)
{

```

```

case '0':tempi=tempi*10;tempi=tempi+0;break;
case 1:tempi=tempi*10;tempi=tempi+1;break;
case 2:tempi=tempi*10;tempi=tempi+2;break;
case 3:tempi=tempi*10;tempi=tempi+3;break;
case 4:tempi=tempi*10;tempi=tempi+4;break;
case 5:tempi=tempi*10;tempi=tempi+5;break;
case 6:tempi=tempi*10;tempi=tempi+6;break;
case 7:tempi=tempi*10;tempi=tempi+7;break;
case 8:tempi=tempi*10;tempi=tempi+8;break;
case 9:tempi=tempi*10;tempi=tempi+9;break;
case 'A':tempi=0;break;
case 'B':jumlah=tempi;tempi=0;lcd_clear();
        lcd_putsf("Ada perubahan harga?");delay_ms(1000);
goto cek4a;break;
}
goto cek3;
//=====
=====
cek4a:
tempc=keypad();
switch(tempc)
{
case 'C':goto cek5;
case 'B':lcd_clear();lcd_putsf("Harga baru:");goto cek4b;
}

delay_ms(10);
goto cek4a;

//=====
=====
cek4b:
sprintf(text,"%5u A-Reset",tempi);
lcd_gotoxy(0,1);
lcd_puts(text);
tempc=keypad();
switch(tempc)
{
case '0':tempi=tempi*10;tempi=tempi+0;break;
case 1:tempi=tempi*10;tempi=tempi+1;break;
case 2:tempi=tempi*10;tempi=tempi+2;break;
case 3:tempi=tempi*10;tempi=tempi+3;break;
case 4:tempi=tempi*10;tempi=tempi+4;break;
case 5:tempi=tempi*10;tempi=tempi+5;break;
case 6:tempi=tempi*10;tempi=tempi+6;break;
case 7:tempi=tempi*10;tempi=tempi+7;break;

```

```

case 8:tempi=tempi*10;tempi=tempi+8;break;
case 9:tempi=tempi*10;tempi=tempi+9;break;
case 'A':tempi=0;break;
case 'B':
switch(pilihan)
{
case kopi:harga_kopi=tempi;break;
case gula:harga_gula=tempi;break;
case beras:harga_beras=tempi;break;
}
tempi=0;goto cek5;break;
}

```

```
goto cek4b;
```

```
//=====
=====
```

```

cek5:
switch(pilihan)
{
case kopi:
lcd_clear();
sprintf(text,"Kopi %5ugr",jumlah);
lcd_puts(text);break;
case gula:
lcd_clear();
sprintf(text,"Gula %5ugr",jumlah);
lcd_puts(text);break;
case beras:
lcd_clear();
sprintf(text,"Beras %5ugr",jumlah);
lcd_puts(text);break;
}
lcd_gotoxy(0,1);
lcd_putsf("Lanjutkan?");

```

```

cek5b:
tempc=keypad();
switch(tempc)
{
case 'C':
lcd_clear();
lcd_putsf("Masukkan pilihan");
delay_ms(1000);
lcd_clear();

```

```
lcd_putsf("1=kopi 2=gula 3=beras");
goto cek1;break;
```

```
case 'B':
switch(pilihan)
{
case kopi:kopi_buka;break;
case gula:gula_buka;break;
case beras:beras_buka;break;
}
goto cek6;break;
}
```

```
goto cek5b;
```

```
//=====
=====
```

```
cek6:
temp1=read_adc(0);
gr=berat(temp1);
switch(pilihan)
{
case kopi:
hargatotal=gr*harga_kopi;break;
case gula:
hargatotal=gr*harga_gula;break;
case beras:
hargatotal=gr*harga_beras;break;
}
```

```
lcd_gotoxy(0,1);
sprintf(text,"%5u gr Rp.%5u",gr,hargatotal);
lcd_puts(text);
```

```
if(gr>=(jumlah))
{
switch(pilihan)
{
case kopi:kopi_tutup;break;
case gula:gula_tutup;break;
case beras:beras_tutup;break;
}
}
```

```
lcd_clear();  
lcd_putsf("Harga Total:");  
goto cek7;  
}
```

```
cek[0]=cek[1];  
cek[1]=gr;
```

```
if(cek[1]<=cek[0])  
cek[2]++;
```

```
if(cek[2]>=25)  
{  
cek[2]=0;  
led_nyala;  
lcd_clear();  
lcd_putsf("Tabung kosong");  
lcd_gotoxy(0,1);  
lcd_putsf("Tekan YA");
```

```
while(keypad()!='B');  
led_mati;  
delay_ms(1);  
switch(pilihan)  
{  
case kopi:kopi_tutup;break;  
case gula:gula_tutup;break;  
case beras:beras_tutup;break;  
}
```

```
lcd_gotoxy(0,1);  
sprintf(text,"%5u gr Rp.%5u",gr,hargatotal);  
lcd_puts(text);  
while(keypad()!='B');  
lcd_clear();  
lcd_putsf("Masukkan pilihan");  
delay_ms(1000);  
lcd_clear();  
lcd_putsf("1=kopi 2=gula 3=beras");  
goto cek1;  
}
```

```

delay_ms(100);
goto cek6;
//=====
=====
cek7:
temp1=read_adc(0);
gr=berat(temp1);

switch(pilihan)
{
case kopi:
hargatotal=gr*harga_kopi;break;
case gula:
hargatotal=gr*harga_gula;break;
case beras:
hargatotal=gr*harga_beras;break;
}

lcd_gotoxy(0,1);
sprintf(text,"%5u gr Rp.%5u",gr,hargatotal);
lcd_puts(text);

tempc=keypad();
if(tempc=='B')
{
lcd_clear();
lcd_putsf("ada pengulangan?");
goto cek8;
}
delay_ms(100);
goto cek7;

//=====
=====
cek8:

tempc=keypad();
if(tempc=='B')
{
switch(pilihan)
{
case kopi:lcd_clear();lcd_putsf("mengulang kopi");kopi_buka;break;
case gula:lcd_clear();lcd_putsf("mengulang gula");gula_buka;break;
case beras:lcd_clear();lcd_putsf("mengulang beras");beras_buka;break;
}
}

```

```
goto cek6;break;
}

if(tempc=='C')
{
lcd_clear();
lcd_putsf("Masukkan pilihan");
delay_ms(1000);
lcd_clear();
lcd_putsf("1=kopi 2=gula 3=beras");
goto cek1;
}

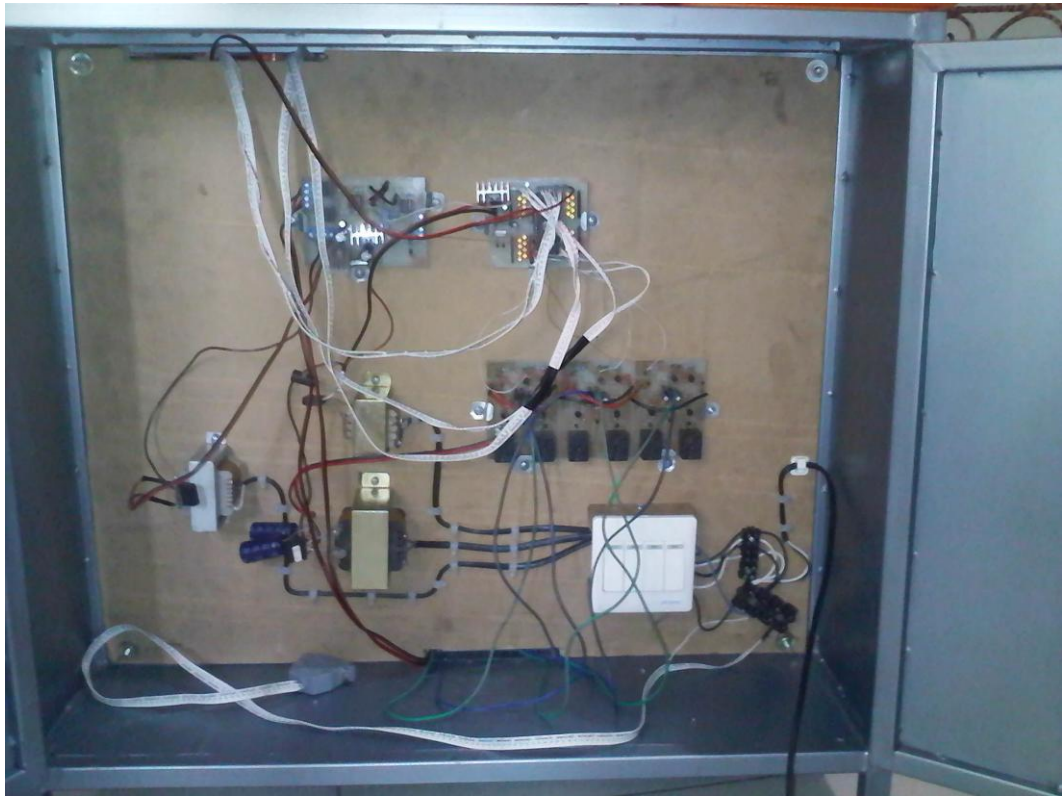
goto cek8;

};
}
```


LAMPIRAN B
GAMBAR SISTEM







LAMPIRAN C
DATASHEET

Features

- High-performance, Low-power AVR[®] 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 16K Bytes of In-System Self-Programmable Flash
 - Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - 512 Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 1K Byte Internal SRAM
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG Interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad MLF
- Operating Voltages
 - 2.7 - 5.5V for ATmega16L
 - 4.5 - 5.5V for ATmega16
- Speed Grades
 - 0 - 8 MHz for ATmega16L
 - 0 - 16 MHz for ATmega16



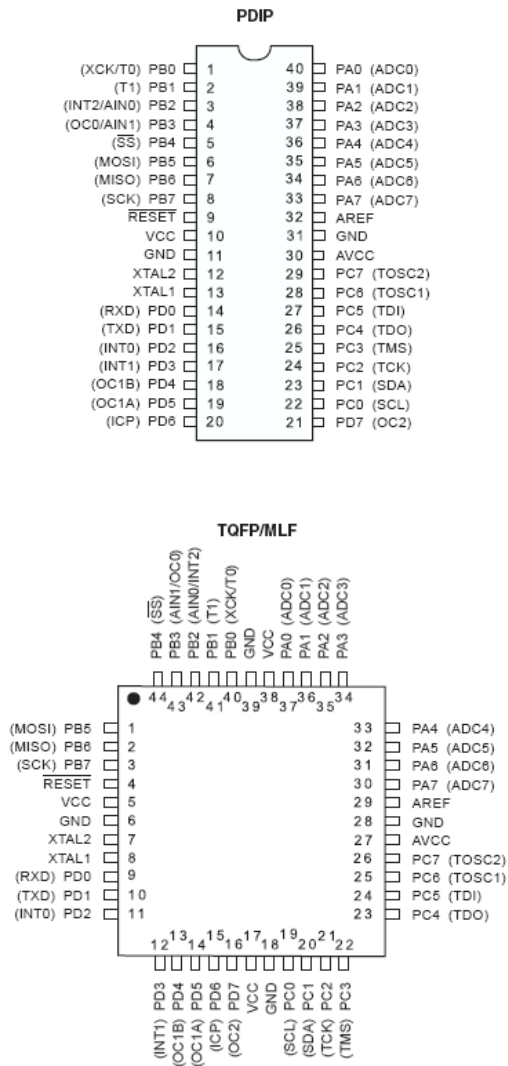
8-bit AVR[®]
Microcontroller
with 16K Bytes
In-System
Programmable
Flash

ATmega16
ATmega16L

Preliminary

Pin Configurations

Figure 1. Pinouts ATmega16



Disclaimer

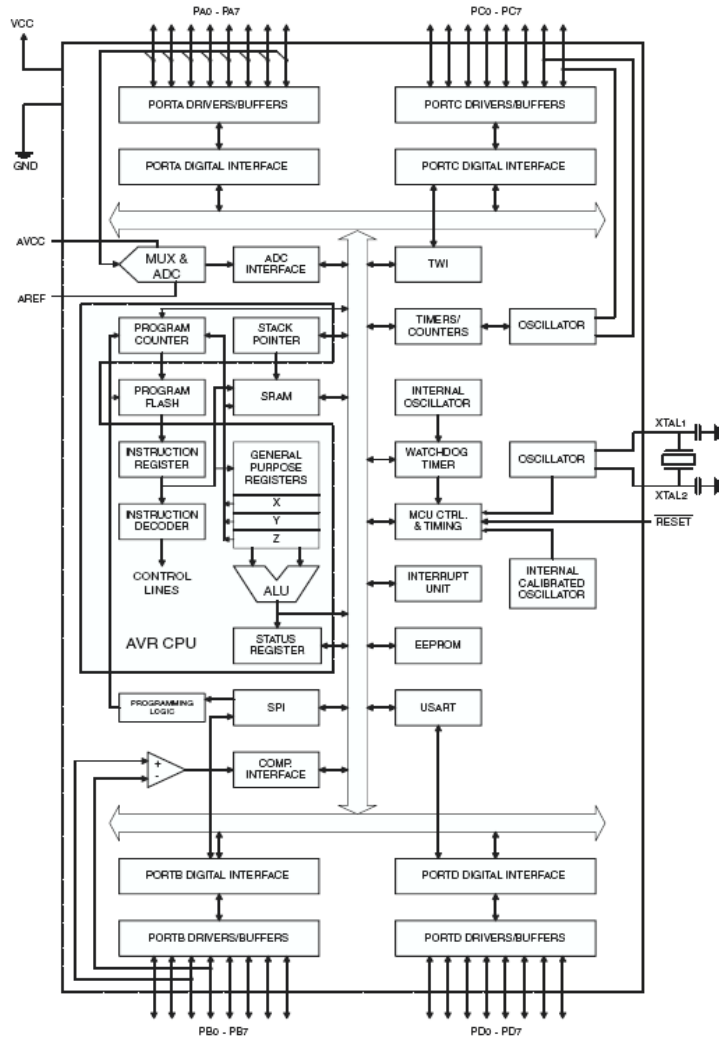
Typical values contained in this data sheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

Overview

The ATmega16 is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega16 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

Block Diagram

Figure 2. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega16 provides the following features: 16K bytes of In-System Programmable Flash Program memory with Read-While-Write capabilities, 512 bytes EEPROM, 1K byte SRAM, 32 general purpose I/O lines, 32 general purpose working registers, a JTAG interface for Boundary-scan, On-chip Debugging support and programming, three flexible Timer/Counters with compare modes, Internal and External Interrupts, a serial programmable USART, a byte oriented Two-wire Serial Interface, an 8-channel, 10-bit ADC with optional differential input stage with programmable gain (TQFP package only), a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the USART, Two-wire interface, A/D Converter, SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next External Interrupt or Hardware Reset. In Power-save mode, the Asynchronous Timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except Asynchronous Timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low-power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run.

The device is manufactured using Atmel's high density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega16 is a powerful microcontroller that provides a highly-flexible and cost-effective solution to many embedded control applications.

The ATmega16 AVR is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC	Digital supply voltage.
GND	Ground.
Port A (PA7..PA0)	Port A serves as the analog inputs to the A/D Converter. Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7..PB0)	<p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega16 as listed on page 55.</p>
Port C (PC7..PC0)	<p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.</p> <p>Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 58.</p>
Port D (PD7..PD0)	<p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega16 as listed on page 60.</p>
RESET	<p>Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 35. Shorter pulses are not guaranteed to generate a reset.</p>
XTAL1	<p>Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.</p>
XTAL2	<p>Output from the inverting Oscillator amplifier.</p>
AVCC	<p>AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.</p>
AREF	<p>AREF is the analog reference pin for the A/D Converter.</p>

About Code Examples

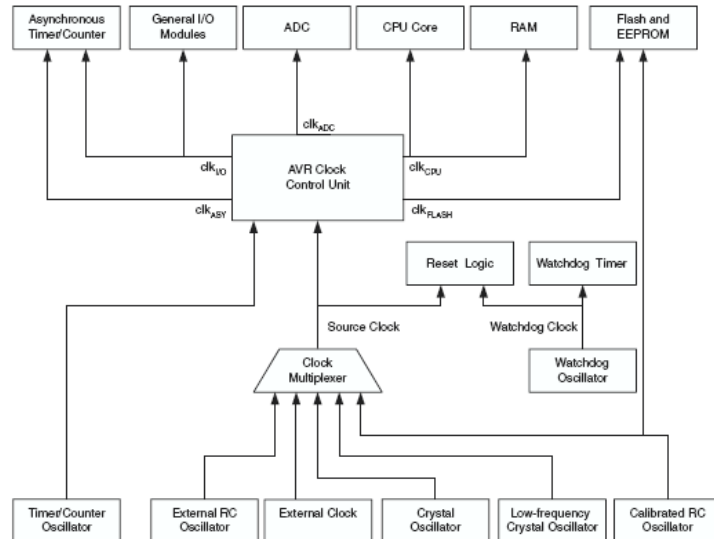
This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C Compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C Compiler documentation for more details.

System Clock and Clock Options

Clock Systems and their Distribution

Figure 11 presents the principal clock systems in the AVR and their distribution. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes, as described in "Power Management and Sleep Modes" on page 30. The clock systems are detailed Figure 11.

Figure 11. Clock Distribution



CPU Clock – clk_{CPU}

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

I/O Clock – $clk_{I/O}$

The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted. Also note that address recognition in the TWI module is carried out asynchronously when $clk_{I/O}$ is halted, enabling TWI address reception in all sleep modes.

Flash Clock – clk_{FLASH}

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

Asynchronous Timer Clock – clk_{ASY} The Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external 32 kHz clock crystal. The dedicated clock domain allows using this Timer/Counter as a real-time counter even when the device is in sleep mode.

ADC Clock – clk_{ADC} The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results.

Clock Sources The device has the following clock source options, selectable by Flash Fuse bits as shown below. The clock from the selected source is input to the AVR clock generator, and routed to the appropriate modules.

Table 2. Device Clocking Options Select⁽¹⁾

Device Clocking Option	CKSEL3..0
External Crystal/Ceramic Resonator	1111 - 1010
External Low-frequency Crystal	1001
External RC Oscillator	1000 - 0101
Calibrated Internal RC Oscillator	0100 - 0001
External Clock	0000

Note: 1. For all fuses "1" means unprogrammed while "0" means programmed.

The various choices for each clocking option is given in the following sections. When the CPU wakes up from Power-down or Power-save, the selected clock source is used to time the start-up, ensuring stable Oscillator operation before instruction execution starts. When the CPU starts from Reset, there is as an additional delay allowing the power to reach a stable level before commencing normal operation. The Watchdog Oscillator is used for timing this real-time part of the start-up time. The number of WDT Oscillator cycles used for each time-out is shown in Table 3. The frequency of the Watchdog Oscillator is voltage dependent as shown in "ATmega16 Typical Characteristics – Preliminary Data" on page 293. The device is shipped with CKSEL = "0001" and SUT = "10" (1 MHz Internal RC Oscillator, slowly rising power).

Table 3. Number of Watchdog Oscillator Cycles

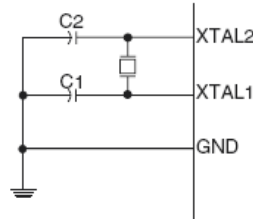
Typ Time-out ($V_{CC} = 5.0V$)	Typ Time-out ($V_{CC} = 3.0V$)	Number of Cycles
4.1 ms	4.3 ms	4K (4,096)
65 ms	69 ms	64K (65,536)

Crystal Oscillator XTAL1 and XTAL2 are input and output, respectively, of an inverting amplifier which can be configured for use as an On-chip Oscillator, as shown in Figure 12. Either a quartz crystal or a ceramic resonator may be used. The CKOPT Fuse selects between two different Oscillator amplifier modes. When CKOPT is programmed, the Oscillator output will oscillate with a full rail-to-rail swing on the output. This mode is suitable when operating in a very noisy environment or when the output from XTAL2 drives a second clock buffer. This mode has a wide frequency range. When CKOPT is unprogrammed, the Oscillator has a smaller output swing. This reduces power consumption considerably. This mode has a limited frequency range and it can not be used to drive other clock buffers.

For resonators, the maximum frequency is 8 MHz with CKOPT unprogrammed and 16 MHz with CKOPT programmed. C1 and C2 should always be equal for both crystals

and resonators. The optimal value of the capacitors depends on the crystal or resonator in use, the amount of stray capacitance, and the electromagnetic noise of the environment. Some initial guidelines for choosing capacitors for use with crystals are given in Table 4. For ceramic resonators, the capacitor values given by the manufacturer should be used. For more information on how to choose capacitors and other details on Oscillator operation, refer to the Multi-purpose Oscillator application note.

Figure 12. Crystal Oscillator Connections



The Oscillator can operate in three different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3..1 as shown in Table 4.

Table 4. Crystal Oscillator Operating Modes

CKOPT	CKSEL3..1	Frequency Range (MHz)	Recommended Range for Capacitors C1 and C2 for Use with Crystals (pF)
1	101 ⁽¹⁾	0.4 - 0.9	–
1	110	0.9 - 3.0	12 - 22
1	111	3.0 - 8.0	12 - 22
0	101, 110, 111	1.0 ≤	12 - 22

Note: 1. This option should not be used with crystals, only with ceramic resonators.

The CKSEL0 Fuse together with the SUT1..0 fuses select the start-up times as shown in Table 5.

Table 5. Start-up Times for the Crystal Oscillator Clock Selection

CKSEL0	SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V _{CC} = 5.0V)	Recommended Usage
0	00	258 CK ⁽¹⁾	4.1 ms	Ceramic resonator, fast rising power
0	01	258 CK ⁽¹⁾	65 ms	Ceramic resonator, slowly rising power
0	10	1K CK ⁽²⁾	–	Ceramic resonator, BOD enabled
0	11	1K CK ⁽²⁾	4.1 ms	Ceramic resonator, fast rising power
1	00	1K CK ⁽²⁾	65 ms	Ceramic resonator, slowly rising power
1	01	16K CK	–	Crystal Oscillator, BOD enabled
1	10	16K CK	4.1 ms	Crystal Oscillator, fast rising power
1	11	16K CK	65 ms	Crystal Oscillator, slowly rising power

Notes: 1. These options should only be used when not operating close to the maximum frequency of the device, and only if frequency stability at start-up is not important for the application. These options are not suitable for crystals.
 2. These options are intended for use with ceramic resonators and will ensure frequency stability at start-up. They can also be used with crystals when not operating close to the maximum frequency of the device, and if frequency stability at start-up is not important for the application.

Low-frequency Crystal Oscillator

To use a 32.768 kHz watch crystal as the clock source for the device, the Low-frequency Crystal Oscillator must be selected by setting the CKSEL fuses to "1001". The crystal should be connected as shown in Figure 12. By programming the CKOPT Fuse, the user can enable internal capacitors on XTAL1 and XTAL2, thereby removing the need for external capacitors. The internal capacitors have a nominal value of 36 pF. Refer to the 32 kHz Crystal Oscillator application note for details on Oscillator operation and how to choose appropriate values for C1 and C2.

When this Oscillator is selected, start-up times are determined by the SUT fuses as shown in Table 6.

Table 6. Start-up Times for the Low-frequency Crystal Oscillator Clock Selection

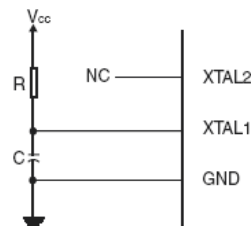
SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	Recommended Usage
00	1K CK ⁽¹⁾	4.1 ms	Fast rising power or BOD enabled
01	1K CK ⁽¹⁾	65 ms	Slowly rising power
10	32K CK	65 ms	Stable frequency at start-up
11	Reserved		

Note: 1. These options should only be used if frequency stability at start-up is not important for the application.

External RC Oscillator

For timing insensitive applications, the external RC configuration shown in Figure 13 can be used. The frequency is roughly estimated by the equation $f = 1/(3RC)$. C should be at least 22 pF. By programming the CKOPT Fuse, the user can enable an internal 36 pF capacitor between XTAL1 and GND, thereby removing the need for an external capacitor. For more information on Oscillator operation and details on how to choose R and C, refer to the External RC Oscillator application note.

Figure 13. External RC Configuration



The Oscillator can operate in four different modes, each optimized for a specific frequency range. The operating mode is selected by the fuses CKSEL3..0 as shown in Table 7.

Table 7. External RC Oscillator Operating Modes

CKSEL3..0	Frequency Range (MHz)
0101	≤ 0.9
0110	0.9 - 3.0
0111	3.0 - 8.0
1000	8.0 - 12.0

When this Oscillator is selected, start-up times are determined by the SUT fuses as shown in Table 8.

Table 8. Start-up Times for the External RC Oscillator Clock Selection

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset (V _{CC} = 5.0V)	Recommended Usage
00	18 CK	–	BOD enabled
01	18 CK	4.1 ms	Fast rising power
10	18 CK	65 ms	Slowly rising power
11	6 CK ⁽¹⁾	4.1 ms	Fast rising power or BOD enabled

Note: 1. This option should not be used when operating close to the maximum frequency of the device.

Calibrated Internal RC Oscillator

The Calibrated Internal RC Oscillator provides a fixed 1.0, 2.0, 4.0, or 8.0 MHz clock. All frequencies are nominal values at 5V and 25°C. This clock may be selected as the system clock by programming the CKSEL fuses as shown in Table 9. If selected, it will operate with no external components. The CKOPT Fuse should always be unprogrammed when using this clock option. During Reset, hardware loads the calibration byte into the OSCCAL Register and thereby automatically calibrates the RC Oscillator. At 5V, 25°C and 1.0 MHz Oscillator frequency selected, this calibration gives a frequency within ± 1% of the nominal frequency. When this Oscillator is used as the Chip Clock, the Watchdog Oscillator will still be used for the Watchdog Timer and for the reset time-out. For more information on the pre-programmed calibration value, see the section "Calibration Byte" on page 256.

Table 9. Internal Calibrated RC Oscillator Operating Modes

CKSEL3..0	Nominal Frequency (MHz)
0001 ⁽¹⁾	1.0
0010	2.0
0011	4.0
0100	8.0

Note: 1. The device is shipped with this option selected.

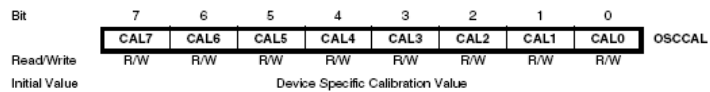
When this Oscillator is selected, start-up times are determined by the SUT fuses as shown in Table 10. XTAL1 and XTAL2 should be left unconnected (NC).

Table 10. Start-up Times for the Internal Calibrated RC Oscillator Clock Selection

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	Recommended Usage
00	6 CK	–	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10 ⁽¹⁾	6 CK	65 ms	Slowly rising power
11	Reserved		

Note: 1. The device is shipped with this option selected.

Oscillator Calibration Register – OSCCAL



• **Bits 7..0 – CAL7..0: Oscillator Calibration Value**

Writing the calibration byte to this address will trim the Internal Oscillator to remove process variations from the Oscillator frequency. This is done automatically during Chip Reset. When OSCCAL is zero, the lowest available frequency is chosen. Writing non-zero values to this register will increase the frequency of the Internal Oscillator. Writing \$FF to the register gives the highest available frequency. The calibrated Oscillator is used to time EEPROM and Flash access. If EEPROM or Flash is written, do not calibrate to more than 10% above the nominal frequency. Otherwise, the EEPROM or Flash write may fail. Note that the Oscillator is intended for calibration to 1.0, 2.0, 4.0, or 8.0 MHz. Tuning to other values is not guaranteed, as indicated in Table 11.

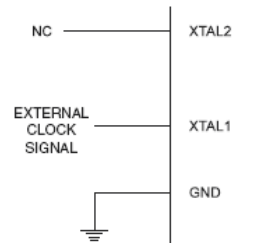
Table 11. Internal RC Oscillator Frequency Range.

OSCCAL Value	Min Frequency in Percentage of Nominal Frequency (%)	Max Frequency in Percentage of Nominal Frequency (%)
\$00	50	100
\$7F	75	150
\$FF	100	200

External Clock

To drive the device from an external clock source, XTAL1 should be driven as shown in Figure 14. To run the device on an external clock, the CKSEL fuses must be programmed to "0000". By programming the CKOPT Fuse, the user can enable an internal 36 pF capacitor between XTAL1 and GND.

Figure 14. External Clock Drive Configuration



When this clock source is selected, start-up times are determined by the SUT fuses as shown in Table 12.

Table 12. Start-up Times for the External Clock Selection

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	Recommended Usage
00	6 CK	–	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10	6 CK	65 ms	Slowly rising power
11	Reserved		

Timer/Counter Oscillator

For AVR microcontrollers with Timer/Counter Oscillator pins (TOSC1 and TOSC2), the crystal is connected directly between the pins. No external capacitors are needed. The Oscillator is optimized for use with a 32.768 kHz watch crystal. Applying an external clock source to TOSC1 is not recommended.

System Control and Reset

Resetting the AVR

During Reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. The instruction placed at the Reset Vector must be a JMP – absolute jump – instruction to the reset handling routine. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa. The circuit diagram in Figure 15 shows the reset logic. Table 15 defines the electrical parameters of the reset circuitry.

The I/O ports of the AVR are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

After all reset sources have gone inactive, a delay counter is invoked, stretching the Internal Reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the CKSEL Fuses. The different selections for the delay period are presented in "Clock Sources" on page 23.

Reset Sources

The ATmega16 has five sources of reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POT}).
- External Reset. The MCU is reset when a low level is present on the \overline{RESET} pin for longer than the minimum pulse length.
- Watchdog Reset. The MCU is reset when the Watchdog Timer period expires and the Watchdog is enabled.
- Brown-out Reset. The MCU is reset when the supply voltage V_{CC} is below the Brown-out Reset threshold (V_{BOT}) and the Brown-out Detector is enabled.
- JTAG AVR Reset. The MCU is reset as long as there is a logic one in the Reset Register, one of the scan chains of the JTAG system. Refer to the section "IEEE 1149.1 (JTAG) Boundary-scan" on page 222 for details.

Figure 15. Reset Logic

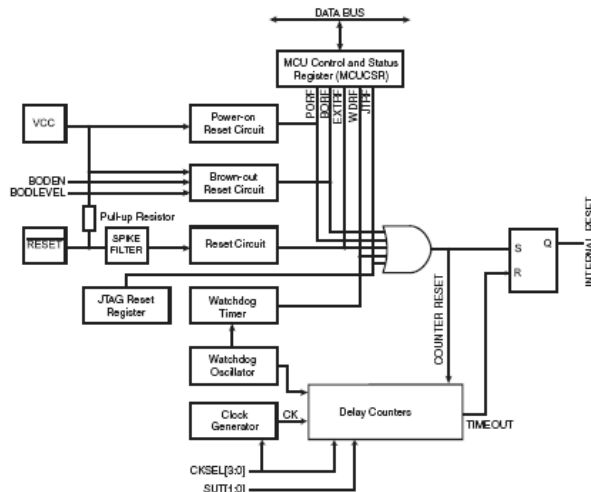


Table 15. Reset Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V _{POT}	Power-on Reset Threshold Voltage (rising)			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling) ⁽¹⁾			1.3	2.3	V
V _{RST}	RESET Pin Threshold Voltage		0.2 V _{CC}		0.85V _{CC}	V
t _{RST}	Minimum pulse width on RESET Pin			50		ns
V _{BOT}	Brown-out Reset Threshold Voltage ⁽²⁾	BODLEVEL = 1	2.5	2.7	3.2	V
		BODLEVEL = 0	3.7	4.0	4.2	
t _{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1		2		μs
		BODLEVEL = 0		2		μs
V _{HYST}	Brown-out Detector hysteresis			50		mV

Notes: 1. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling).
 2. V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to V_{CC} = V_{BOT} during the production test. This guarantees that a Brown-out Reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 1 for ATmega16L and BODLEVEL = 0 for ATmega16. BODLEVEL = 1 is not applicable for ATmega16.

Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip detection circuit. The detection level is defined in Table 15. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the Start-up Reset, as well as to detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after V_{CC} rise. The RESET signal is activated again, without any delay, when V_{CC} decreases below the detection level.

Figure 16. MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC} .

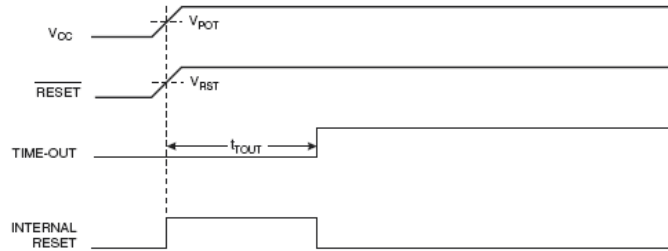
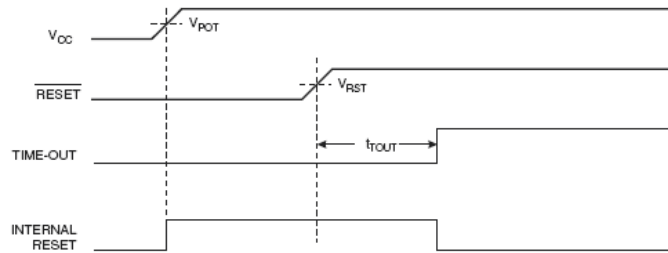


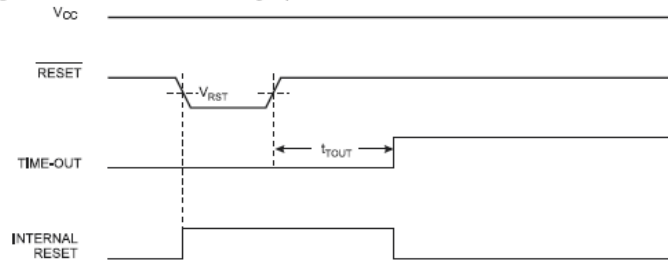
Figure 17. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



External Reset

An External Reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than the minimum pulse width (see Table 15) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage - V_{RST} - on its positive edge, the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

Figure 18. External Reset During Operation



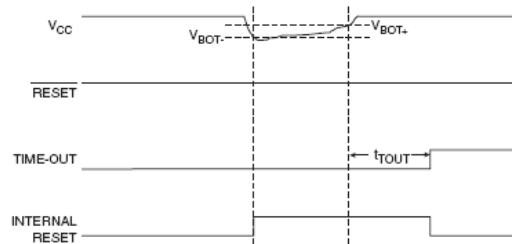
Brown-out Detection

ATmega16 has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST}/2$ and $V_{BOT-} = V_{BOT} - V_{HYST}/2$.

The BOD circuit can be enabled/disabled by the fuse BODEN. When the BOD is enabled (BODEN programmed), and V_{CC} decreases to a value below the trigger level (V_{BOT} , in Figure 19), the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level (V_{BOT+} , in Figure 19), the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than t_{BOD} given in Table 15.

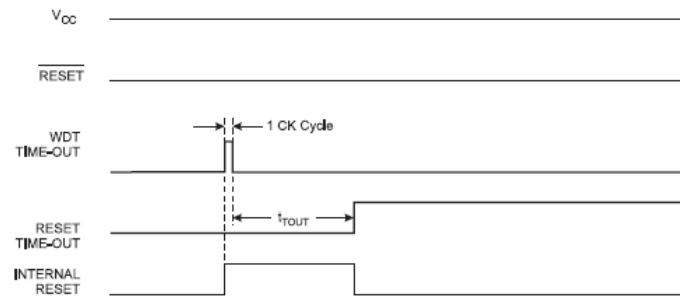
Figure 19. Brown-out Reset During Operation



Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT} . Refer to page 39 for details on operation of the Watchdog Timer.

Figure 20. Watchdog Reset During Operation

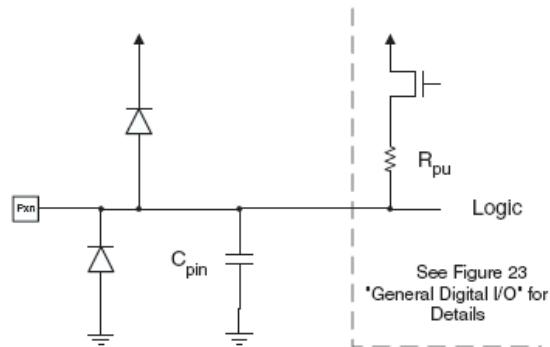


I/O Ports

Introduction

All AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both high sink and source capability. The pin driver is strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both V_{CC} and Ground as indicated in Figure 22. Refer to "Electrical Characteristics" on page 285 for a complete list of parameters.

Figure 22. I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case "x" represents the numbering letter for the port, and a lower case "n" represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. i.e., PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in "Register Description for I/O Ports" on page 62.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. In addition, the Pull-up Disable – PUD bit in SFIOR disables the pull-up function for all pins in all ports when set.

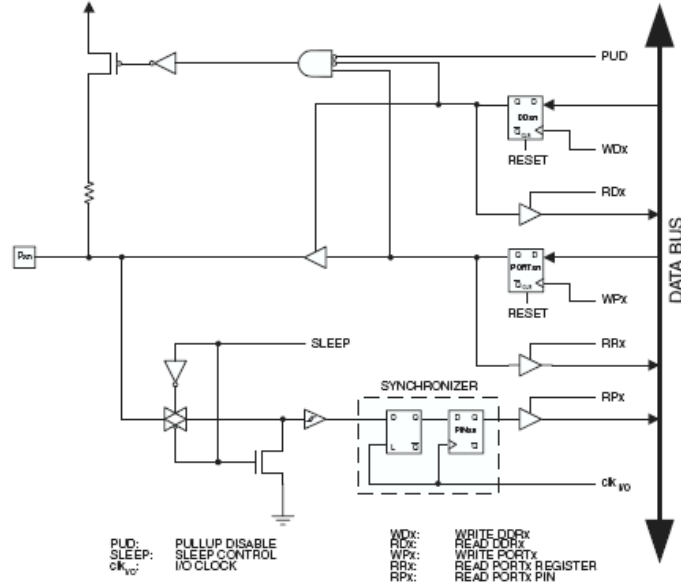
Using the I/O port as General Digital I/O is described in "Ports as General Digital I/O" on page 48. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in "Alternate Port Functions" on page 52. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 23 shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 23. General Digital I/O⁽¹⁾



Note: 1. WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. clk_{IO}, SLEEP, and PUD are common to all ports.

Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. As shown in "Register Description for I/O Ports" on page 62, the DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when a reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

When switching between tri-state ((DDxn, PORTxn) = 0b00) and output high ((DDxn, PORTxn) = 0b11), an intermediate state with either pull-up enabled ((DDxn, PORTxn) = 0b01) or output low ((DDxn, PORTxn) = 0b10) must occur. Normally, the pull-up

enabled state is fully acceptable, as a high-impedant environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the SFIOR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ($\{DDxn, PORTxn\} = 0b00$) or the output high state ($\{DDxn, PORTxn\} = 0b11$) as an intermediate step.

Table 20 summarizes the control signals for the pin value.

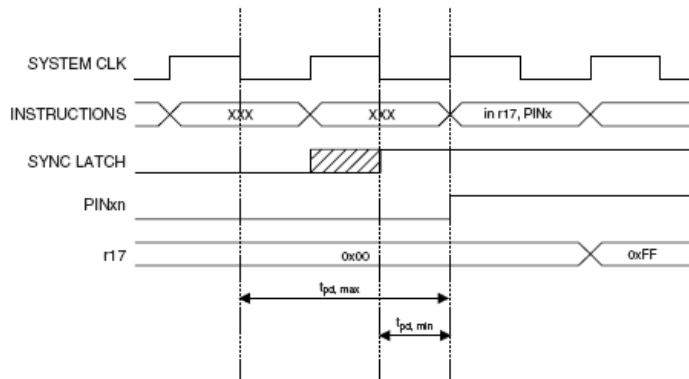
Table 20. Port Pin Configurations

DDxn	PORTxn	PUD (in SFIOR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register bit. As shown in Figure 23, the PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 24 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{pd,max}$ and $t_{pd,min}$ respectively.

Figure 24. Synchronization when Reading an Externally Applied Pin Value

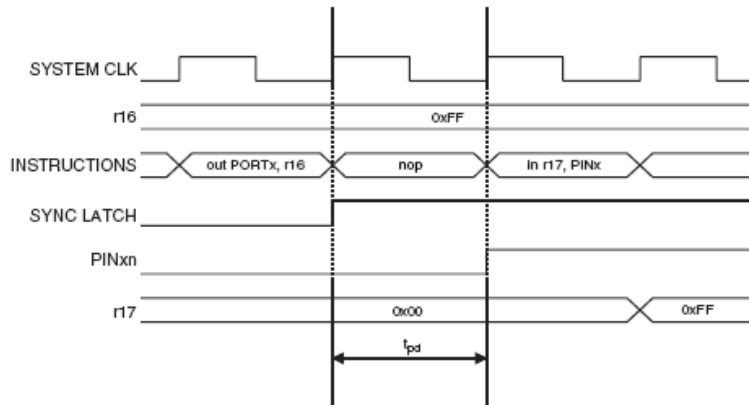


Consider the clock period starting shortly *after* the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the "SYNC LATCH" signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn Register at the

succeeding positive clock edge. As indicated by the two arrows $t_{pd,max}$ and $t_{pd,min}$, a single signal transition on the pin will be delayed between $\frac{1}{2}$ and $1\frac{1}{2}$ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a *nop* instruction must be inserted as indicated in Figure 25. The *out* instruction sets the "SYNC LATCH" signal at the positive edge of the clock. In this case, the delay t_{pd} through the synchronizer is one system clock period.

Figure 25. Synchronization when Reading a Software Assigned Pin Value



External Interrupts

The External Interrupts are triggered by the INT0, INT1, and INT2 pins. Observe that, if enabled, the interrupts will trigger even if the INT0..2 pins are configured as outputs. This feature provides a way of generating a software interrupt. The external interrupts can be triggered by a falling or rising edge or a low level (INT2 is only an edge triggered interrupt). This is set up as indicated in the specification for the MCU Control Register – MCUCR – and MCU Control and Status Register – MCUCSR. When the external interrupt is enabled and is configured as level triggered (only INT0/INT1), the interrupt will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT0 and INT1 requires the presence of an I/O clock, described in “Clock Systems and their Distribution” on page 22. Low level interrupts on INT0/INT1 and the edge interrupt on INT2 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode. The I/O clock is halted in all sleep modes except Idle mode.

Note that if a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by the Watchdog Oscillator clock. The period of the Watchdog Oscillator is 1 μ s (nominal) at 5.0V and 25°C. The frequency of the Watchdog Oscillator is voltage dependent as shown in “Electrical Characteristics” on page 285. The MCU will wake up if the input has the required level during this sampling or if it is held until the end of the start-up time. The start-up time is defined by the SUT fuses as described in “System Clock and Clock Options” on page 22. If the level is sampled twice by the Watchdog Oscillator clock but disappears before the end of the start-up time, the MCU will still wake up, but no interrupt will be generated. The required level must be held long enough for the MCU to complete the wake up to trigger the level interrupt.

MCU Control Register – MCUCR

The MCU Control Register contains control bits for interrupt sense control and general MCU functions.

Bit	7	6	5	4	3	2	1	0	
	SM2	SE	SM1	SM0	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 3, 2 – ISC11, ISC10: Interrupt Sense Control 1 Bit 1 and Bit 0

The External Interrupt 1 is activated by the external pin INT1 if the SREG I-bit and the corresponding interrupt mask in the GICR are set. The level and edges on the external INT1 pin that activate the interrupt are defined in Table 34. The value on the INT1 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 34. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

- Bit 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask are set. The level and edges on the external INT0 pin that activate the interrupt are defined in Table 35. The value on the INT0 pin is sampled before detecting edges. If edge or toggle interrupt is selected, pulses that last longer than one clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt.

Table 35. Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Any logical change on INT0 generates an interrupt request.
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

MCU Control and Status Register – MCUCSR

Bit	7	6	5	4	3	2	1	0	MCUCSR
	JTD	ISC2	-	JTRF	WDRF	BORF	EXTRF	PORF	
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0						See Bit Description

- Bit 6 – ISC2: Interrupt Sense Control 2

The Asynchronous External Interrupt 2 is activated by the external pin INT2 if the SREG I-bit and the corresponding interrupt mask in GICR are set. If ISC2 is written to zero, a falling edge on INT2 activates the interrupt. If ISC2 is written to one, a rising edge on INT2 activates the interrupt. Edges on INT2 are registered asynchronously. Pulses on INT2 wider than the minimum pulse width given in Table 36 will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. When changing the ISC2 bit, an interrupt can occur. Therefore, it is recommended to first disable INT2 by clearing its Interrupt Enable bit in the GICR Register. Then, the ISC2 bit can be changed. Finally, the INT2 Interrupt Flag should be cleared by writing a logical one to its Interrupt Flag bit (INTF2) in the GIFR Register before the interrupt is re-enabled.

Table 36. Asynchronous External Interrupt Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
t_{INT}	Minimum pulse width for asynchronous external interrupt			50		ns

General Interrupt Control Register – GICR

Bit	7	6	5	4	3	2	1	0	GICR
	INT1	INT0	INT2	-	-	-	IVSEL	IVCE	
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- Bit 7 – INT1: External Interrupt Request 1 Enable

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control1 bits 1/0 (ISC11 and

ISC10) in the MCU General Control Register (MCUCR) define whether the External Interrupt is activated on rising and/or falling edge of the INT1 pin or level sensed. Activity on the pin will cause an interrupt request even if INT1 is configured as an output. The corresponding interrupt of External Interrupt Request 1 is executed from the INT1 interrupt Vector.

- **Bit 6 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU General Control Register (MCUCR) define whether the External Interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. Activity on the pin will cause an interrupt request even if INT0 is configured as an output. The corresponding interrupt of External Interrupt Request 0 is executed from the INT0 interrupt vector.

- **Bit 5 – INT2: External Interrupt Request 2 Enable**

When the INT2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control2 bit (ISC2) in the MCU Control and Status Register (MCUCSR) defines whether the External Interrupt is activated on rising or falling edge of the INT2 pin. Activity on the pin will cause an interrupt request even if INT2 is configured as an output. The corresponding interrupt of External Interrupt Request 2 is executed from the INT2 Interrupt Vector.

General Interrupt Flag Register – GIFR

Bit	7	6	5	4	3	2	1	0	
	INTF1	INTF0	INTF2	-	-	-	-	-	GIFR
Read/Write	R/W	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – INTF1: External Interrupt Flag 1**

When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in GICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

- **Bit 6 – INTF0: External Interrupt Flag 0**

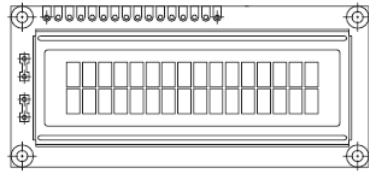
When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in GICR are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

- **Bit 5 – INTF2: External Interrupt Flag 2**

When an event on the INT2 pin triggers an interrupt request, INTF2 becomes set (one). If the I-bit in SREG and the INT2 bit in GICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. Note that when entering some sleep modes with the INT2 interrupt disabled, the input buffer on this pin will be disabled. This may cause a logic change in internal signals which will set the INTF2 flag. See "Digital Input Enable and Sleep Modes" on page 51 for more information.



16 x 2 Character LCD



FEATURES

- 5 x 8 dots with cursor
- Built-in controller (KS 0066 or Equivalent)
- + 5V power supply (Also available for + 3V)
- 1/16 duty cycle
- B/L to be driven by pin 1, pin 2 or pin 15, pin 16 or A.K (LED)
- N.V. optional for + 3V power supply

MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module Dimension	80.0 x 36.0	mm
Viewing Area	66.0 x 16.0	mm
Dot Size	0.56 x 0.66	mm
Character Size	2.96 x 5.56	mm

ABSOLUTE MAXIMUM RATING					
ITEM	SYMBOL	STANDARD VALUE			UNIT
		MIN.	TYP.	MAX.	
Power Supply	VDD-VSS	- 0.3	-	7.0	V
Input Voltage	VI	- 0.3	-	VDD	V

NOTE: VSS = 0 Volt, VDD = 5.0 Volt

ELECTRICAL SPECIFICATIONS							
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT	
			MIN.	TYP.	MAX.		
Input Voltage	VDD	VDD = + 5V	4.7	5.0	5.3	V	
		VDD = + 3V	2.7	3.0	5.3	V	
Supply Current	IDD	VDD = 5V	-	1.2	3.0	mA	
Recommended LC Driving Voltage for Normal Temp. Version Module	VDD - V0	- 20 °C	-	-	-	V	
		0 °C	4.2	4.8	5.1		
		25 °C	3.8	4.2	4.6		
		50 °C	3.6	4.0	4.4		
LED Forward Voltage	VF	25 °C	-	4.2	4.6	V	
LED Forward Current	IF	25 °C	Array	-	130	260	mA
			Edge	-	20	40	
EL Power Supply Current	IEL	Vel = 110VAC:400Hz	-	-	5.0	mA	

DISPLAY CHARACTER ADDRESS CODE:																
Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DD RAM Address	00	01														0F
DD RAM Address	40	41														4F

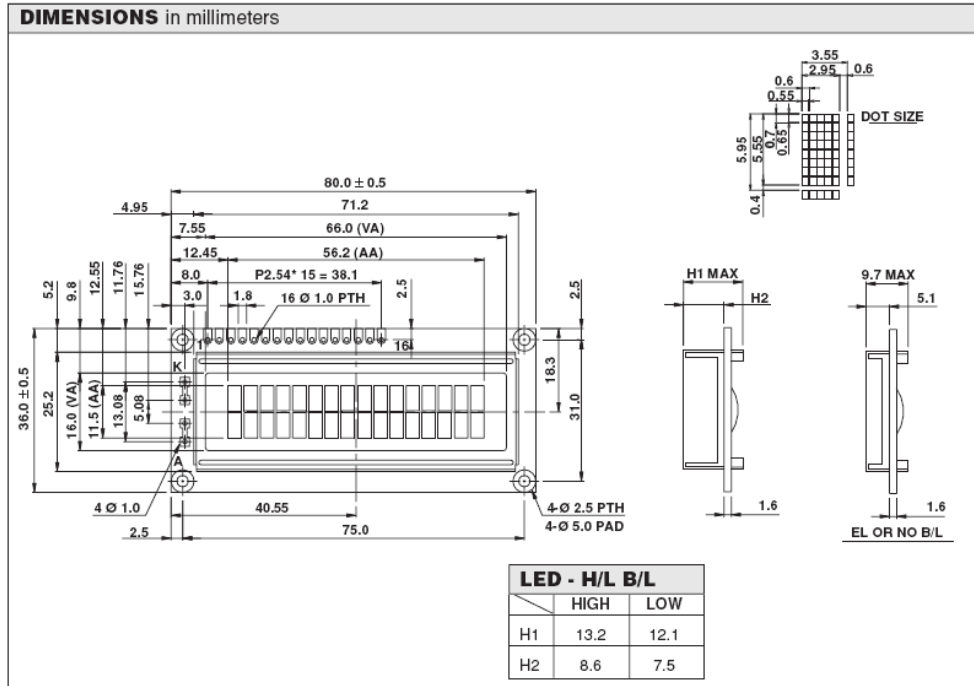
LCD-016M002B

Vishay

16 x 2 Character LCD



PIN NUMBER	SYMBOL	FUNCTION
1	V _{ss}	GND
2	V _{dd}	+ 3V or + 5V
3	V _o	Contrast Adjustment
4	RS	H/L Register Select Signal
5	R/W	H/L Read/Write Signal
6	E	H → L Enable Signal
7	DB0	H/L Data Bus Line
8	DB1	H/L Data Bus Line
9	DB2	H/L Data Bus Line
10	DB3	H/L Data Bus Line
11	DB4	H/L Data Bus Line
12	DB5	H/L Data Bus Line
13	DB6	H/L Data Bus Line
14	DB7	H/L Data Bus Line
15	A/V _{ee}	+ 4.2V for LED/Negative Voltage Output
16	K	Power Supply for B/L (OV)





INA121

FET-Input, Low Power INSTRUMENTATION AMPLIFIER

FEATURES

- **LOW BIAS CURRENT:** $\pm 4\text{pA}$
- **LOW QUIESCENT CURRENT:** $\pm 450\mu\text{A}$
- **LOW INPUT OFFSET VOLTAGE:** $\pm 200\mu\text{V}$
- **LOW INPUT OFFSET DRIFT:** $\pm 2\mu\text{V}/^\circ\text{C}$
- **LOW INPUT NOISE:**
 $20\text{nV}/\sqrt{\text{Hz}}$ at $f = 1\text{kHz}$ ($G = 100$)
- **HIGH CMR:** 106dB
- **WIDE SUPPLY RANGE:** $\pm 2.25\text{V}$ to $\pm 18\text{V}$
- **LOW NONLINEARITY ERROR:** 0.001% max
- **INPUT PROTECTION TO $\pm 40\text{V}$**
- **8-PIN DIP AND SO-8 SURFACE MOUNT**

APPLICATIONS

- **LOW-LEVEL TRANSDUCER AMPLIFIERS**
 Bridge, RTD, Thermocouple
- **PHYSIOLOGICAL AMPLIFIERS**
 ECG, EEG, EMG, Respiratory
- **HIGH IMPEDANCE TRANSDUCERS**
- **CAPACITIVE SENSORS**
- **MULTI-CHANNEL DATA ACQUISITION**
- **PORTABLE, BATTERY OPERATED SYSTEMS**
- **GENERAL PURPOSE INSTRUMENTATION**

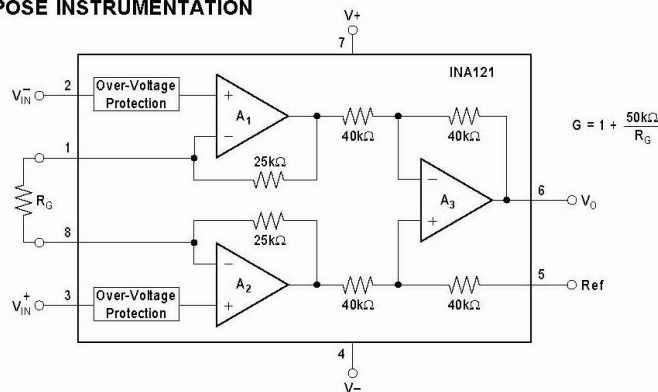
DESCRIPTION

The INA121 is a FET-input, low power instrumentation amplifier offering excellent accuracy. Its versatile three-op amp design and very small size make it ideal for a variety of general purpose applications. Low bias current ($\pm 4\text{pA}$) allows use with high impedance sources.

Gain can be set from 1V to 10,000V/V with a single external resistor. Internal input protection can withstand up to $\pm 40\text{V}$ without damage.

The INA121 is laser-trimmed for very low offset voltage ($\pm 200\mu\text{V}$), low offset drift ($\pm 2\mu\text{V}/^\circ\text{C}$), and high common-mode rejection (106dB at $G = 100$). It operates on power supplies as low as $\pm 2.25\text{V}$ ($+4.5\text{V}$), allowing use in battery operated and single 5V systems. Quiescent current is only $450\mu\text{A}$.

Package options include 8-pin plastic DIP and SO-8 surface mount. All are specified for the -40°C to $+85^\circ\text{C}$ industrial temperature range.



International Airport Industrial Park • Mailing Address: PO Box 11400, Tucson, AZ 85734 • Street Address: 6730 S. Tucson Blvd., Tucson, AZ 85706 • Tel: (520) 746-1111 • Twx: 910-952-1111
 Internet: <http://www.burr-brown.com/> • FAX Line: (800) 548-6133 (US/Canada Only) • Cable: BBRCORP • Telex: 066-6491 • FAX: (520) 889-1510 • Immediate Product Info: (800) 548-6132

NPN general purpose transistors

BC546; BC547

FEATURES

- Low current (max. 100 mA)
- Low voltage (max. 65 V).

APPLICATIONS

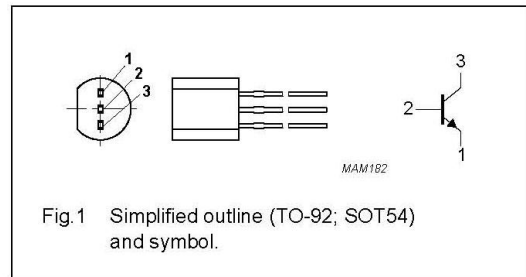
- General purpose switching and amplification.

DESCRIPTION

NPN transistor in a TO-92; SOT54 plastic package.
PNP complements: BC556 and BC557.

PINNING

PIN	DESCRIPTION
1	emitter
2	base
3	collector



LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V _{CBO}	collector-base voltage	open emitter			
	BC546		–	80	V
	BC547		–	50	V
V _{CEO}	collector-emitter voltage	open base			
	BC546		–	65	V
	BC547		–	45	V
V _{EBO}	emitter-base voltage	open collector			
	BC546		–	6	V
	BC547		–	6	V
I _C	collector current (DC)		–	100	mA
I _{CM}	peak collector current		–	200	mA
I _{BM}	peak base current		–	200	mA
P _{tot}	total power dissipation	T _{amb} ≤ 25 °C; note 1	–	500	mW
T _{stg}	storage temperature		–65	+150	°C
T _j	junction temperature		–	150	°C
T _{amb}	operating ambient temperature		–65	+150	°C

Note

1. Transistor mounted on an FR4 printed-circuit board.

KA78XX/KA78XXA

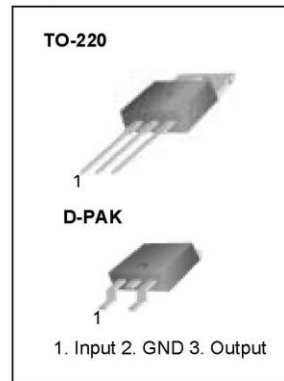
3-Terminal 1A Positive Voltage Regulator

Features

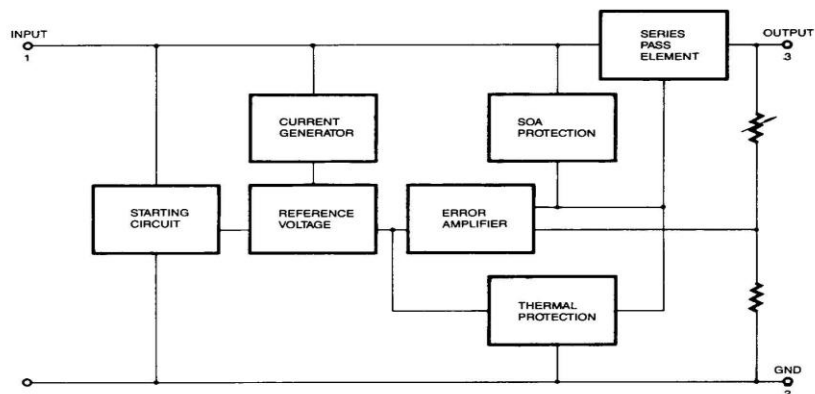
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The KA78XX/KA78XXA series of three-terminal positive regulator are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Diagram



SG7900A/SG7900 SERIES

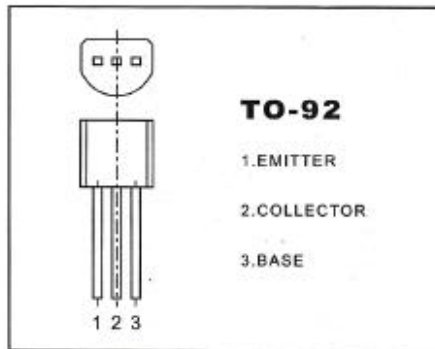
NEGATIVE REGULATOR

CONNECTION DIAGRAMS & ORDERING INFORMATION (See Notes Below)			
Package	Part No.	Ambient Temperature Range	Connection Diagram
3-TERMINAL TO-3 METAL CAN K-PACKAGE	SG79XXAK/883B	-55°C to 125°C	
	SG7905AK/DESC	-55°C to 125°C	
	SG7912AK/DESC	-55°C to 125°C	
	SG7915AK/DESC	-55°C to 125°C	
	SG79XXAK	-55°C to 125°C	
	SG79XXK/883B	-55°C to 125°C	
	JAN7905K	-55°C to 125°C	
	JAN7912K	-55°C to 125°C	
	JAN7915K	-55°C to 125°C	
	SG79XXK	-55°C to 125°C	
SG79XXK	0°C to 125°C		
3-PIN TO-39 METAL CAN T-PACKAGE	SG79XXAT/883B	-55°C to 125°C	
	SG7905AT/DESC	-55°C to 125°C	
	SG7912AT/DESC	-55°C to 125°C	
	SG7915AT/DESC	-55°C to 125°C	
	SG79XXAT	-55°C to 125°C	
	SG79XXT/883B	-55°C to 125°C	
	JAN7905T	-55°C to 125°C	
	JAN7912T	-55°C to 125°C	
	JAN7915T	-55°C to 125°C	
	SG79XXT	-55°C to 125°C	
3-PIN HERMETIC TO-257 IG-PACKAGE (Isolated)	SG79XXAIG/883B	-55°C to 125°C	
	SG7905AIG/DESC	-55°C to 125°C	
	SG7912AIG/DESC	-55°C to 125°C	
	SG7915AIG/DESC	-55°C to 125°C	
	SG79XXAIG	-55°C to 125°C	
	SG79XXIG/883B	-55°C to 125°C	
	SG79XXIG	-55°C to 125°C	
20-PIN CERAMIC LEADLESS CHIP CARRIER L-PACKAGE	SG79XXL/883B	-55°C to 125°C	<p>(See Notes 5 & 6)</p>
	SG79XXL	-55°C to 125°C	
	SG7905AL/DESC	-55°C to 125°C	
	SG7912AL/DESC	-55°C to 125°C	
	SG7915AL/DESC	-55°C to 125°C	

- Note 1. Contact factory for JAN and DESC product availability.
 2. All parts are viewed from the top.
 3. "XX" to be replaced by output voltage of specific fixed regulator.
 4. Some products will be available in hermetic flat pack (F). Consult factory for price and availability.
 5. Both inputs and outputs must be externally connected together at the device terminals.
 6. For normal operation, the V_o SENSE pin must be externally connected to the load.

TO-92 Plastic-Encapsulate Transistors

C945 TRANSISTOR(NPN)



FEATURES

Power dissipation

P_{CM} : 0.4W ($T_{amb}=25^{\circ}C$)

Collector current

I_{CM} : 0.15 A

Collector-base voltage

$V_{(BR)CBO}$: 60 V

Operating and storage junction temperature range

T_j, T_{stg} : $-55^{\circ}C$ to $+150^{\circ}C$

ELECTRICAL CHARACTERISTICS

($T_{amb}=25^{\circ}C$ unless otherwise specified)

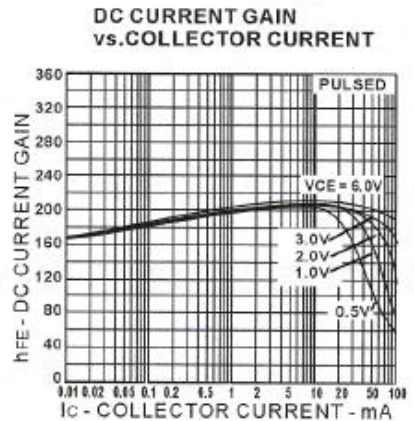
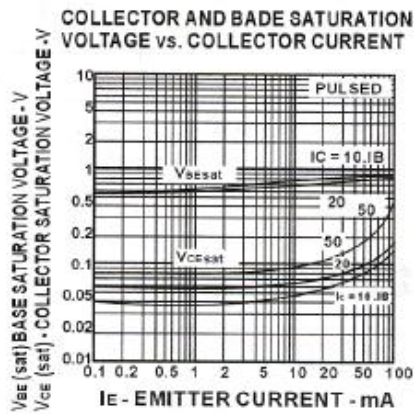
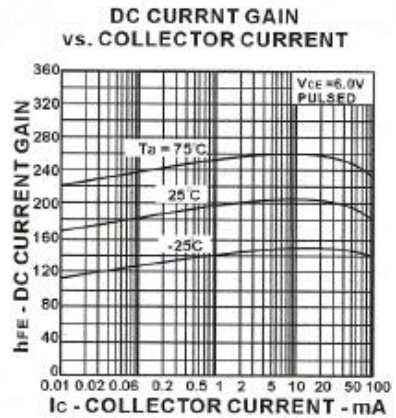
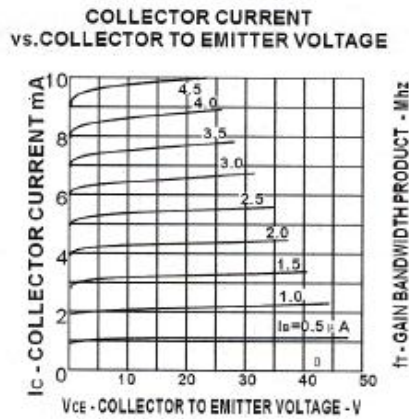
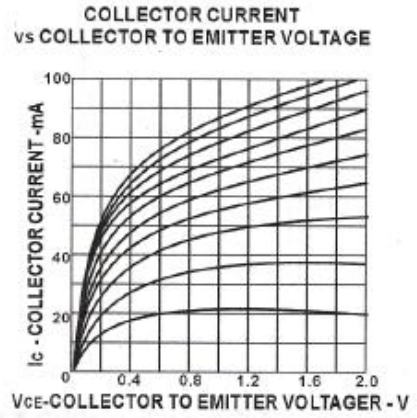
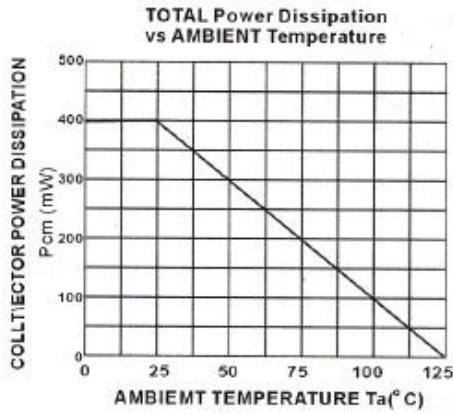
Parameter	Symbol	Test conditions	MIN	MAX	UNIT
Collector-base breakdown voltage	$V_{(BR)CBO}$	$I_C=1000 \mu A, I_E=0$	60		V
Collector-emitter breakdown voltage	$V_{(BR)CEO}$	$I_C=0.1 mA, I_E=0$	50		V
Emitter-base breakdown voltage	$V_{(BR)EBO}$	$I_E=100 \mu A, I_C=0$	5		V
Collector cut-off current	I_{CBO}	$V_{CE}=60 V, I_E=0$		0.1	μA
Collector cut-off current	I_{CER}	$V_{CE}=55 V, R=10 M\Omega$		0.1	μA
Emitter cut-off current	I_{EBO}	$V_{EB}=5 V, I_C=0$		0.1	μA
DC current gain	$h_{FE(1)}$	$V_{CE}=6 V, I_C=1 mA$	70	700	
	$h_{FE(2)}$	$V_{CE}=6 V, I_C=0.1 mA$	40		
Collector-emitter saturation voltage	V_{CEsat}	$I_C=100 mA, I_E=10 mA$		0.3	V
Base-emitter saturation voltage	V_{BEsat}	$I_C=100 mA, I_E=10 mA$		1	V
Base-emitter voltage	V_{BE}	$I_E=310mA$		1.4	V
Transition frequency	f_T	$V_{CE}=6 V, I_C=10 mA$ $f=30MHz$	150		MHz

CLASSIFICATION OF $h_{FE(1)}$

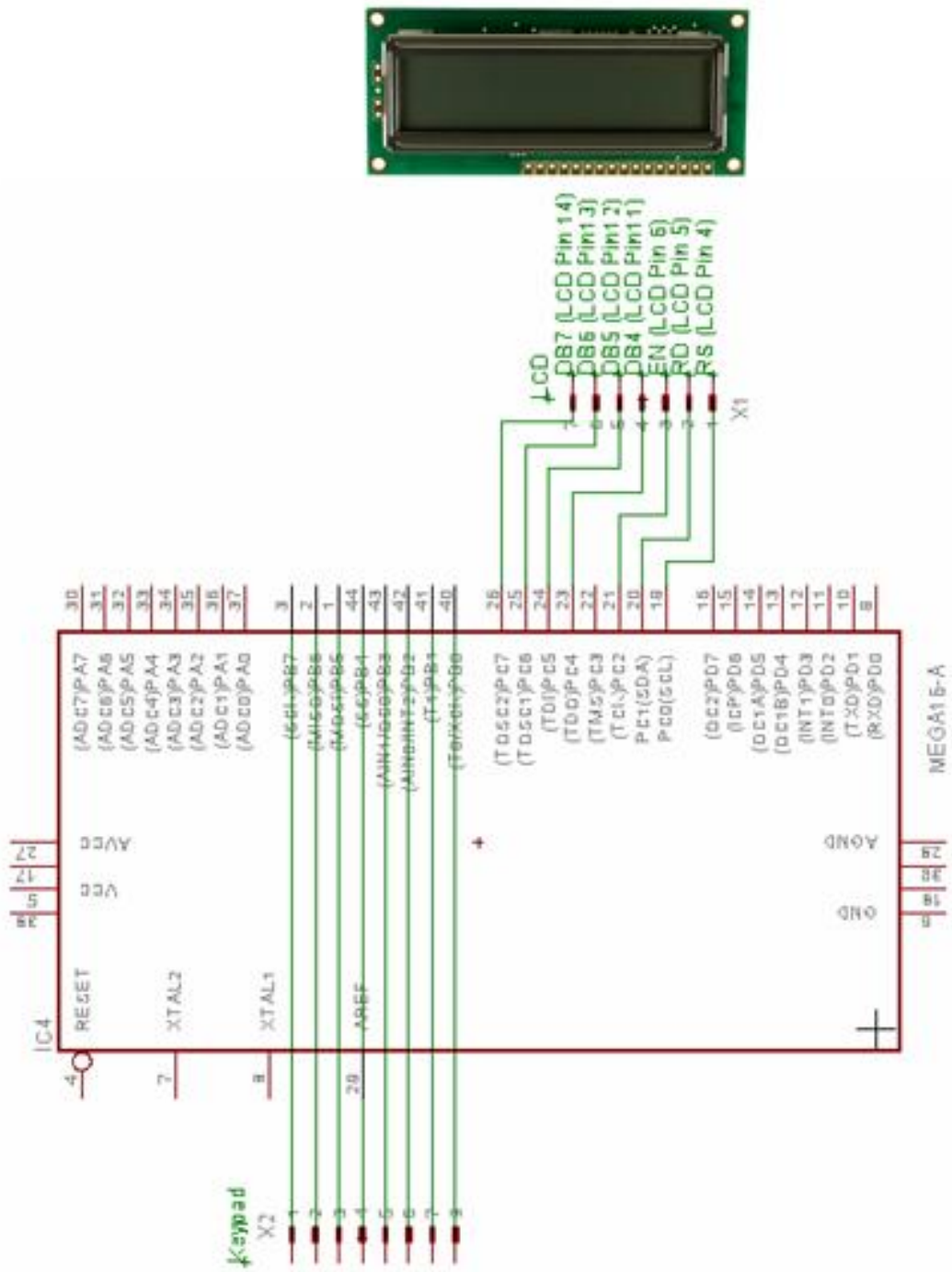
Rank	O	Y	GR	BL
Range	70-140	120-240	200-400	350-700

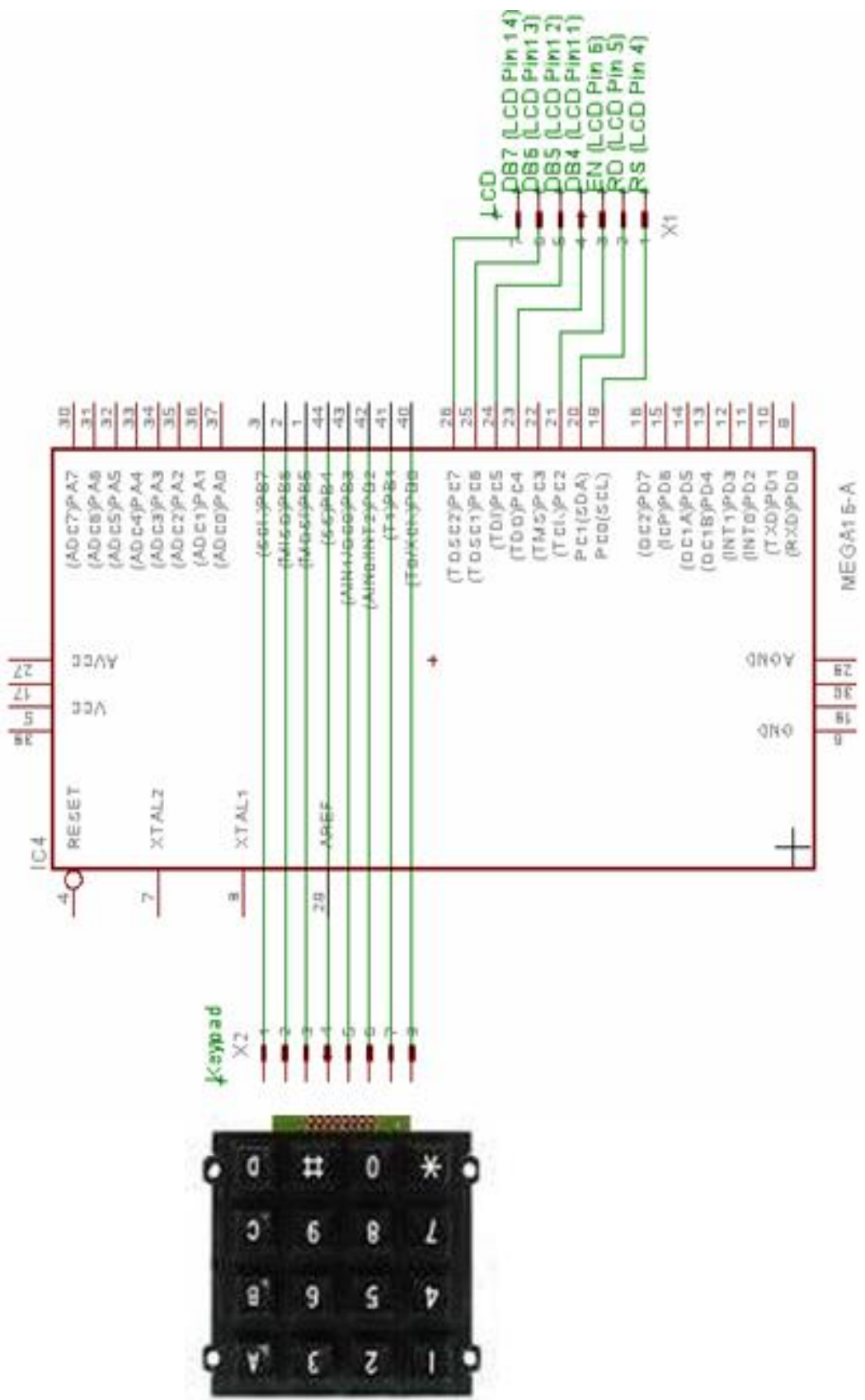
Typical Characteristics

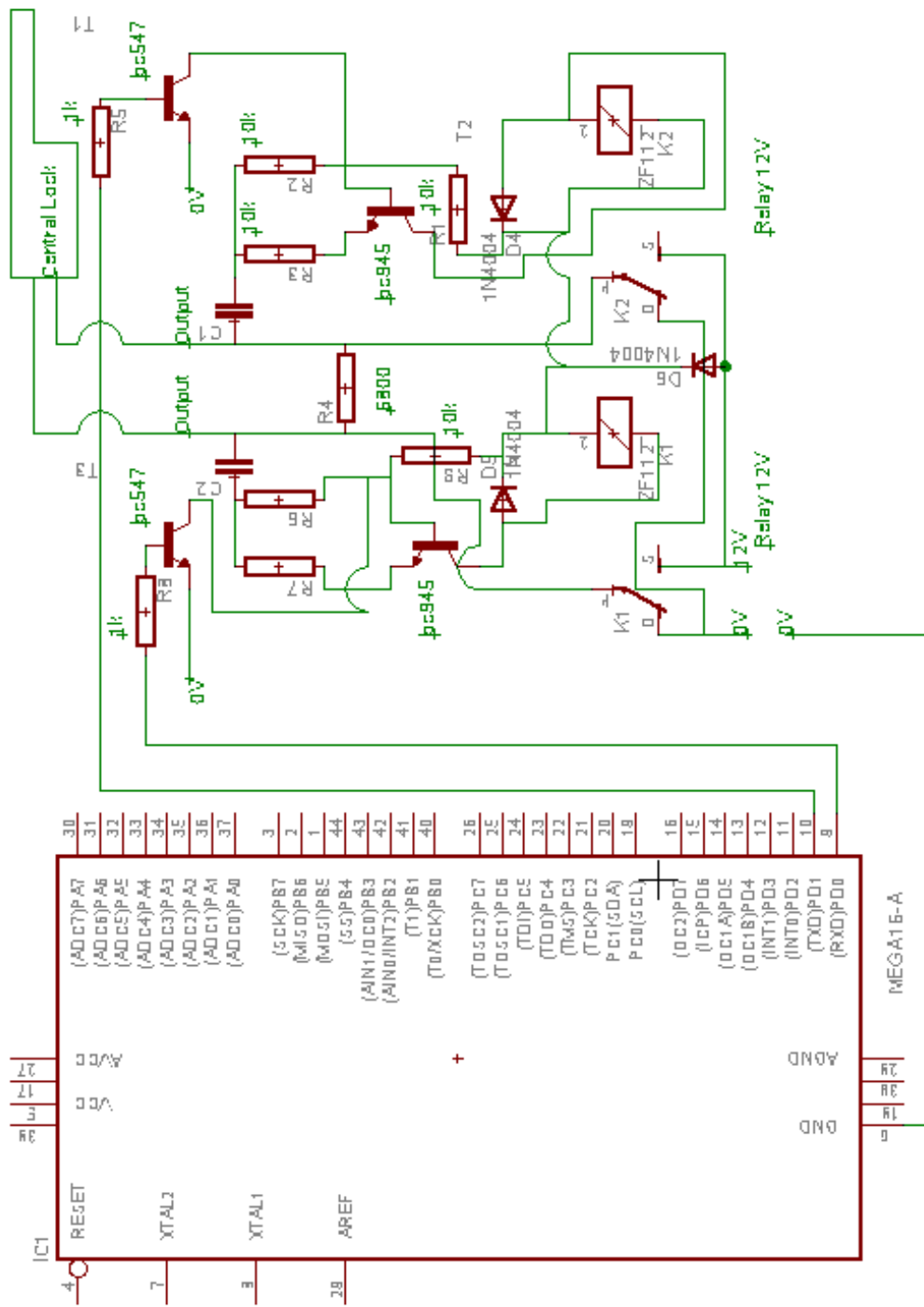
C945

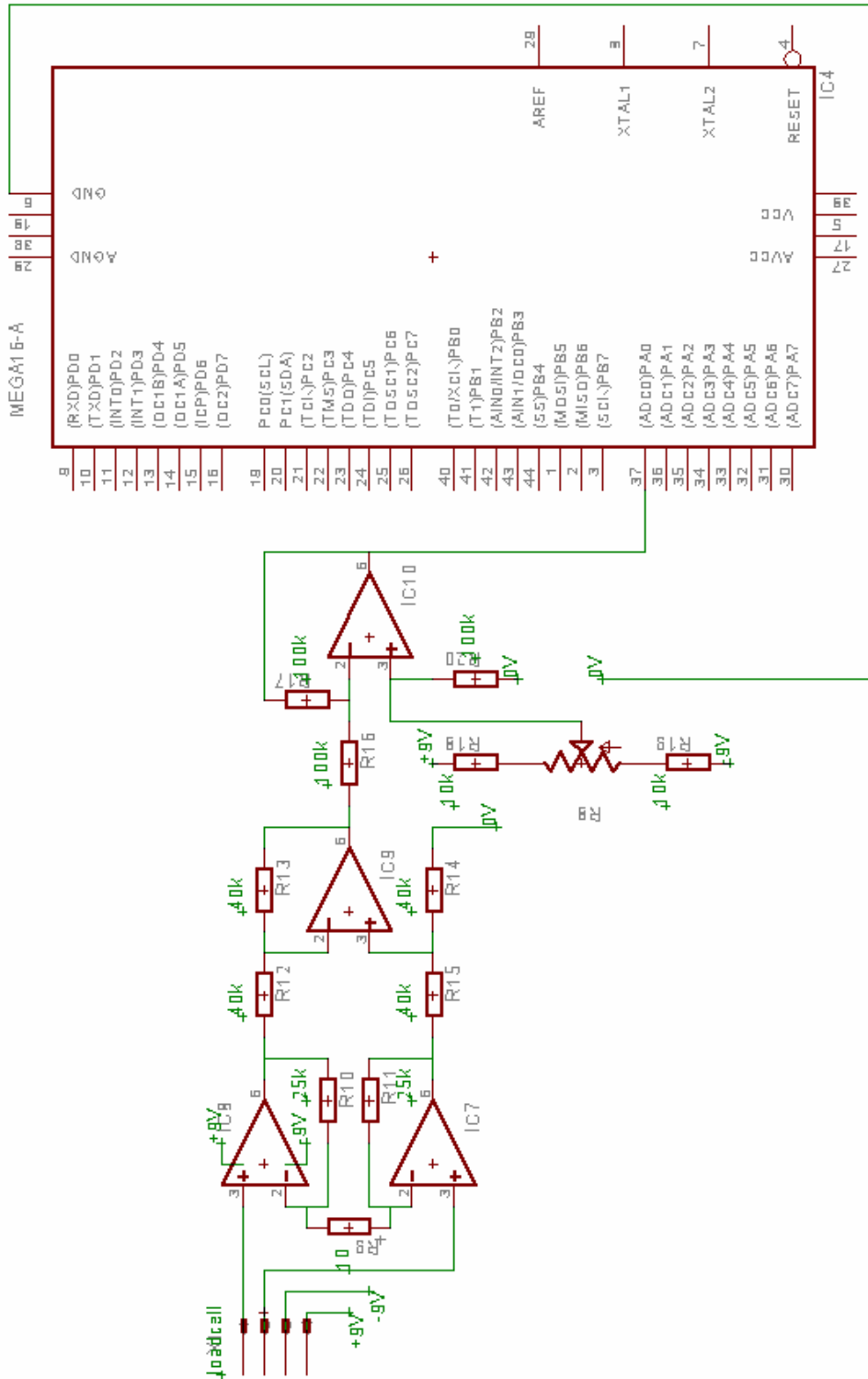


LAMPIRAN D
SKEMATIK RANGKAIAN









LAMPIRAN E
KODE-KODE TOMBOL KEYPAD

