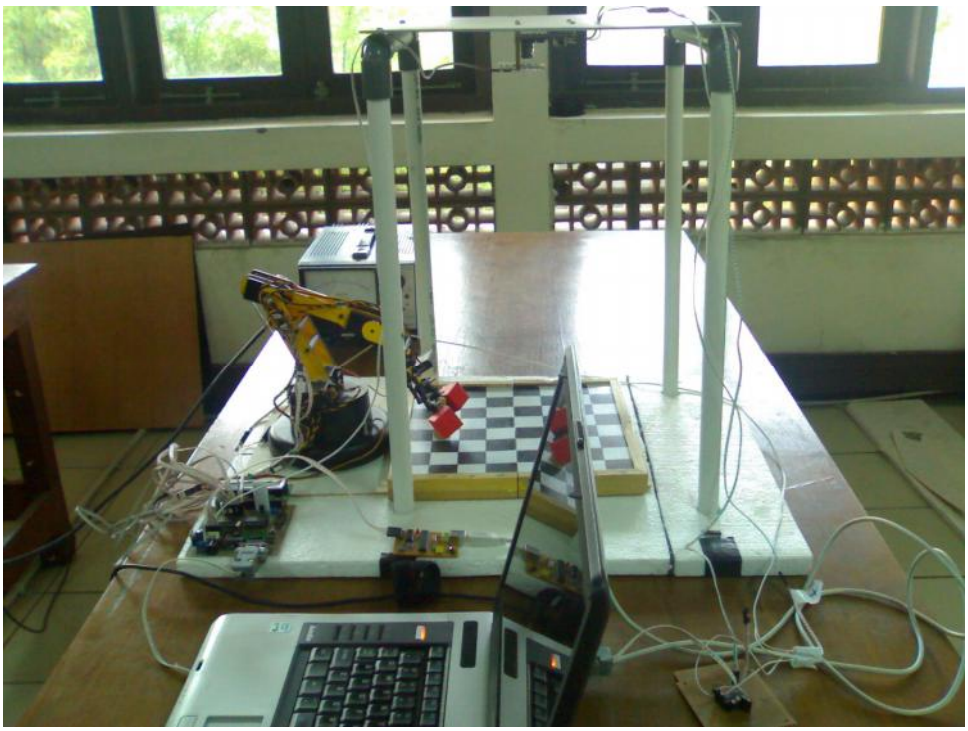
LAMPIRAN - A

FOTO ALAT

TAMPAK SAMPING

TAMPAK DEPAN



LAMPIRAN - B

PROGRAM PENGONTROL MIKRO

PROGRAM VISUAL BASIC

PROGRAM PENGONTROL MIKRO

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

Project : MIMI
Version : 1
Date    : 7/13/2009
Author  : F4CG
Company : F4CG
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 <stdio.h>
#include <delay.h>

int i,mi;
unsigned char ko[32],koor[2];

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

#define RXB8 1
#define TXB8 0
#define UPE 2
#define OVR 3
#define FE 4
#define UDRE 5
#define RXC 7

#define FRAMING_ERROR (1<<FE)
#define PARITY_ERROR (1<<UPE)
#define DATA_OVERRUN (1<<OVR)
#define DATA_REGISTER_EMPTY (1<<UDRE)
#define RX_COMPLETE (1<<RXC)

// USART Receiver buffer
#define RX_BUFFER_SIZE 8
char rx_buffer[RX_BUFFER_SIZE];

#if RX_BUFFER_SIZE<256
unsigned char rx_wr_index,rx_rd_index,rx_counter;
#else
unsigned int rx_wr_index,rx_rd_index,rx_counter;
#endif

// This flag is set on USART Receiver buffer overflow
bit rx_buffer_overflow;

// USART Receiver interrupt service routine
interrupt [USART_RXC] void usart_rx_isr(void)
{
char status,data;
status=UCSRA;
data=UDR;
if ((status & (FRAMING_ERROR | PARITY_ERROR |
DATA_OVERRUN))==0)
   {
   rx_buffer[rx_wr_index]=data;
   if (++rx_wr_index == RX_BUFFER_SIZE)
rx_wr_index=0;
   if (++rx_counter == RX_BUFFER_SIZE)
      {
      rx_counter=0;
      rx_buffer_overflow=1;
      };
   };
   koor[mi]=data;
   if(mi<1) mi++;
   else mi=0;
}

#ifndef _DEBUG_TERMINAL_IO_
// Get a character from the USART Receiver buffer
#define _ALTERNATE_GETCHAR_
#pragma used+
char getchar(void)
{
char data;
while (rx_counter==0);
data=rx_buffer[rx_rd_index];
if (++rx_rd_index == RX_BUFFER_SIZE)
rx_rd_index=0;
#asm("cli")
--rx_counter;
#asm("sei")
return data;
}
#pragma used-
#endif

// Standard Input/Output functions
#include <stdio.h>

// Declare your global variables here

//====SUB PROGRAM KOORDINAT TANGAN

void a (void)
{
//koordinat (0,0)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1900);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    // buka
{
PORTA.6=1;
delay_us(1400);
```

```
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)   //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)   // posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1700);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2200);
PORTA.3=0;
PORTA.4=1;
delay_us(1400);
PORTA.4=0;
}

for(i=0;i<200;i++)  //  posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1700);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2200);
PORTA.3=0;
PORTA.4=1;
delay_us(1300);
PORTA.4=0;
PORTA.6=1;
delay_us(1000);       // tutup
PORTA.6=0;
delay_ms(14);
}
}

void b (void)
{
//koordinat (1,0)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1800);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1800);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2200);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1800);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2200);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void c (void)
{
// koordinat (2,0)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1670);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)   //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
```

```
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(2000);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2300);
PORTA.3=0;
PORTA.4=1;
delay_us(1150);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(2000);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2300);
PORTA.3=0;
PORTA.4=1;
delay_us(1150);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void d (void)
{
//koordinat (3,0)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1500);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(2100);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2300);
PORTA.3=0;
PORTA.4=1;
delay_us(1050);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(2100);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2300);
PORTA.3=0;
PORTA.4=1;
delay_us(1050);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void e (void)
{
//koordinat (4,0)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1300);
PORTA.0=0;
delay_ms(19);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(2000);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
```

```
delay_us(2250);
PORTA.3=0;
PORTA.4=1;
delay_us(1050);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(2000);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2250);
PORTA.3=0;
PORTA.4=1;
delay_us(1050);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void f (void)
{
//koordinat (5,0)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1200);
PORTA.0=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1900);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2250);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1900);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2250);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void g (void)
{
//koordinat (6,0)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1100);
PORTA.0=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1800);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2300);
PORTA.3=0;
PORTA.4=1;
delay_us(1400);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
```

```
PORTA.2=1;
delay_us(1800);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2300);
PORTA.3=0;
PORTA.4=1;
delay_us(1400);
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void h (void)
{
//koordinat (7,0)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1000);
PORTA.0=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)   //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1750);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2250);
PORTA.3=0;
PORTA.4=1;
delay_us(1450);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1750);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2250);
PORTA.3=0;
PORTA.4=1;
delay_us(1450);
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void j (void)
{
//koordinat (0,1)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1830);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)   //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1550);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2200);
PORTA.3=0;
PORTA.4=1;
delay_us(1600);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1550);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2200);
PORTA.3=0;
PORTA.4=1;
delay_us(1600);
PORTA.4=0;
```

```
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void k (void)
{
//koordinat (1,1)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1700);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1600);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2050);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1600);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2050);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void l (void)
{
//koordinat (2,1)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1600);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1650);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2100);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1650);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2100);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void m (void)
{
//koordinat (3,1)
```

```
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1480);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1700);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2130);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1700);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2130);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void n (void)
{
//koordinat (4,1)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1350);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1750);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2200);
PORTA.3=0;
PORTA.4=1;
delay_us(1250);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1750);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2200);
PORTA.3=0;
PORTA.4=1;
delay_us(1250);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}
void o (void)
{
//koordinat (5,1)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1250);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
```

```
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1700);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2150);
PORTA.3=0;
PORTA.4=1;
delay_us(1250);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1700);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2150);
PORTA.3=0;
PORTA.4=1;
delay_us(1250);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void p (void)
{
//koordinat (6,1)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1150);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1600);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2100);
PORTA.3=0;
PORTA.4=1;
delay_us(1300);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1600);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2100);
PORTA.3=0;
PORTA.4=1;
delay_us(1300);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void q (void)
{
//koordinat (7,1)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1050);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
```

```
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2000);
PORTA.3=0;
PORTA.4=1;
delay_us(1300);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2000);
PORTA.3=0;
PORTA.4=1;
delay_us(1300);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void r (void)
{
//koordinat (0,2)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1770);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2000);
PORTA.3=0;
PORTA.4=1;
delay_us(1400);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2000);
PORTA.3=0;
PORTA.4=1;
delay_us(1400);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void s (void)
{
// koordinat (1,2)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1680);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
```

```
PORTA.3=1;
delay_us(2000);
PORTA.3=0;
PORTA.4=1;
delay_us(1350);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2000);
PORTA.3=0;
PORTA.4=1;
delay_us(1350);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void t (void)
{
//koordinat (2,2)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1600);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1950);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1950);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void u (void)
{
//koordinat (3,2)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1500);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1950);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
```

```
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1950);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void v (void)
{
//koordinat (4,2)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1400);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)   //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1950);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1950);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void w (void)
{
//koordinat (5,2)
   for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1280);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)   //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1950);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1950);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
```

```
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void x (void)
{
//koordinat (6,2)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1200);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2100);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1500);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2100);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void y (void)
{
//koordinat (7,2)
for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1130);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1430);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2050);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1430);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(2050);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void z (void)
{
```

```
//koordinat (0,3)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1720);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1350);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1300);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1350);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1300);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void aa (void)
{
//koordinat (1,3)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1650);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1400);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1300);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1400);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1300);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ab (void)
{
//koordinat (2,3)
for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1580);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
```

```
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ac (void)
{
// koordinat (3,3)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1500);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ad (void)
{
// koordinat (4,3)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1400);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
```

```
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ae (void)
{
// koordinat (5,3)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1330);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1200);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void af (void)
{
// koordinat (6,3)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1250);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
```

```
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1250);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1450);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1250);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ag (void)
{
// koordinat (7,3)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1150);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1350);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1250);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1350);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1850);
PORTA.3=0;
PORTA.4=1;
delay_us(1250);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ah (void)
{
// koordinat (0,4)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1700);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1250);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1750);
```

```
PORTA.3=0;
PORTA.4=1;
delay_us(1500);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1250);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1750);
PORTA.3=0;
PORTA.4=1;
delay_us(1500);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ai (void)
{
// koordinat (1,4)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1630);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1300);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1830);
PORTA.3=0;
PORTA.4=1;
delay_us(1500);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1300);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1830);
PORTA.3=0;
PORTA.4=1;
delay_us(1500);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void aj (void)
{
// koordinat (2,4)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1550);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1300);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1830);
PORTA.3=0;
PORTA.4=1;
delay_us(1500);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
```

```
PORTA.1=1;
PORTA.2=1;
delay_us(1300);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1830);
PORTA.3=0;
PORTA.4=1;
delay_us(1500);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ak (void)
{
// koordinat (3,4)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1510);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1330);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1840);
PORTA.3=0;
PORTA.4=1;
delay_us(1510);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
 PORTA.1=1;
PORTA.2=1;
delay_us(1330);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1840);
PORTA.3=0;
PORTA.4=1;
delay_us(1510);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void al (void)
{
// koordinat (4,4)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1430);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1330);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1840);
PORTA.3=0;
PORTA.4=1;
delay_us(1510);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1330);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1840);
PORTA.3=0;
```

```
PORTA.4=1;
delay_us(1510);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void am (void)
{
// koordinat (5,4)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1330);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1330);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1840);
PORTA.3=0;
PORTA.4=1;
delay_us(1510);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1330);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1840);
PORTA.3=0;
PORTA.4=1;
delay_us(1510);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void an (void)
{
// koordinat (6,4)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1300);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1350);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1860);
PORTA.3=0;
PORTA.4=1;
delay_us(1530);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1350);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1860);
PORTA.3=0;
PORTA.4=1;
delay_us(1530);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
```

```
}

void ao (void)
{
// koordinat (7,4)

 for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1200);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1250);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1760);
PORTA.3=0;
PORTA.4=1;
delay_us(1530);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1250);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1760);
PORTA.3=0;
PORTA.4=1;
delay_us(1530);
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ap (void)
{
// koordinat (0,5)

 for(i=0;i<50;i++)    //bawah
{
PORTA.0=1;
delay_us(1700);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)    //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1150);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1700);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1150);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1700);
PORTA.4=0;
PORTA.6=1;            //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void aq (void)
{
// koordinat (1,5)

 for(i=0;i<50;i++)    //bawah
{
```

```
PORTA.0=1;
delay_us(1650);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1600);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1600);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void ar (void)
{
// koordinat (2,5)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1580);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1600);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1600);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void as (void)
{
// koordinat (3,5)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1500);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
```

```
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1800);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1800);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void at (void)
{
// koordinat (4,5)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1440);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)     //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
}

for(i=0;i<200;i++)     //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
PORTA.6=1;             //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void au (void)
{
// koordinat (5,5)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1380);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
```

```
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void av (void)
{
// koordinat (6,5)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1300);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1200);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void aw (void)
{
// koordinat (7,5)

 for(i=0;i<50;i++)     //bawah
{
PORTA.0=1;
delay_us(1270);
PORTA.0=0;
delay_ms(18);
}

for(i=0;i<50;i++)     //buka
{
PORTA.6=1;
delay_us(1400);
PORTA.6=0;
delay_ms(19);
}

for(i=0;i<50;i++)    //lengan
{
PORTA.5=1;
delay_us(1500);
PORTA.5=0;
delay_ms(18);
}

for(i=0;i<200;i++)      //posisi awal
{
```

```
PORTA.1=1;
PORTA.2=1;
delay_us(1150);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
}

for(i=0;i<200;i++)      //posisi awal+tutup
{
PORTA.1=1;
PORTA.2=1;
delay_us(1150);
PORTA.1=0;
PORTA.2=0;
PORTA.3=1;
delay_us(1800);
PORTA.3=0;
PORTA.4=1;
delay_us(1550);
PORTA.4=0;
PORTA.6=1;              //tutup
delay_us(1000);
PORTA.6=0;
delay_ms(14);
}
}

void a1 (void)
{
//koordinat (0,0)
   for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(300);
    PORTA.4=0;
    delay_us(200);
    PORTA.5=0;
    delay_us(200);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(200);
    PORTA.0=0;
    delay_us(300);
    PORTA.3=0;
    delay_ms(17);
    }
}

void b1 (void)
{
//koordinat (1,0)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(300);
    PORTA.5=0;
    delay_us(300);
    PORTA.1=0;
    PORTA.2=0;
    PORTA.0=0;
    delay_us(400);
    PORTA.3=0;
    delay_ms(17);
    }
}

void c1 (void)
{
// koordinat (2,0)
   for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(150);
    PORTA.4=0;
    delay_us(350);
    PORTA.5=0;
    delay_us(170);
    PORTA.0=0;
    delay_us(330);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(300);
    PORTA.3=0;
    delay_ms(17);
    }
}

void d1 (void)
{
//koordinat (3,0)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(50);
    PORTA.4=0;
    delay_us(450);
    PORTA.5=0;
    PORTA.0=0;
    delay_us(600);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(200);
    PORTA.3=0;
    delay_ms(17);
    }
}

void e1 (void)
{
//koordinat (4,0)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(900);
    PORTA.6=0;
    delay_us(150);
    PORTA.4=0;
    delay_us(250);
```

```
    PORTA.0=0;
    delay_us(200);
    PORTA.5=0;
    delay_us(500);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(250);
    PORTA.3=0;
    delay_ms(17);
    }
}

void f1 (void)
{
   //koordinat (5,0)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    PORTA.0=0;
    delay_us(300);
    PORTA.5=0;
    delay_us(400);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(350);
    PORTA.3=0;
    delay_ms(17);
    }
}

void g1 (void)
{
//koordinat (6,0)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(100);
    PORTA.0=0;
    delay_us(300);
    PORTA.4=0;
    delay_us(100);
    PORTA.5=0;
    delay_us(300);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(500);
    PORTA.3=0;
    delay_ms(17);
    }
}

void h1 (void)
{
    //koordinat (7,0)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.0=0;
    PORTA.6=0;
    delay_us(450);
    PORTA.4=0;
    delay_us(50);
    PORTA.5=0;
    delay_us(250);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(500);
    PORTA.3=0;
    delay_ms(17);
    }
}

void j1 (void)
{
//koordinat (0,1)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(500);
    PORTA.5=0;
    delay_us(50);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(50);
    PORTA.4=0;
    delay_us(230);
    PORTA.0=0;
    delay_us(370);
    PORTA.3=0;
    delay_ms(17);
    }
}

void k1 (void)
{
   //koordinat (1,1)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(300);
    PORTA.5=0;
    delay_us(100);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(100);
    PORTA.0=0;
    delay_us(350);
    PORTA.3=0;
    delay_ms(17);
    }
}

void l1 (void)
{
//koordinat (2,1)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
```

```
    PORTA.4=0;
    delay_us(300);
    PORTA.5=0;
    delay_us(100);
    PORTA.0=0;
    delay_us(50);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(450);
    PORTA.3=0;
    delay_ms(17);
    }
}

void m1 (void)
{
//koordinat (3,1)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(280);
    PORTA.0=0;
    delay_us(20);
    PORTA.5=0;
    delay_us(200);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(430);
    PORTA.3=0;
    delay_ms(17);
    }
}

void n1 (void)
{
//koordinat (4,1)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(250);
    PORTA.4=0;
    delay_us(100);
    PORTA.0=0;
    delay_us(150);
    PORTA.5=0;
    delay_us(250);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(450);
    PORTA.3=0;
    delay_ms(17);
    }
}

void o1 (void)
{
//koordinat (5,1)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(250);
    PORTA.4=0;
    PORTA.0=0;
    delay_us(250);
    PORTA.5=0;
    delay_us(200);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(450);
    PORTA.3=0;
    delay_ms(17);
    }
}

void p1 (void)
{
//koordinat (6,1)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(150);
    PORTA.0=0;
    delay_us(150);
    PORTA.4=0;
    delay_us(200);
    PORTA.5=0;
    delay_us(100);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(500);
    PORTA.3=0;
    delay_ms(17);
    }
}

void q1 (void)
{
//koordinat (7,1)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(50);
    PORTA.0=0;
    delay_us(250);
    PORTA.4=0;
    delay_us(200);
    PORTA.5=0;
    PORTA.1=0;
    PORTA.2=0;
    delay_us(500);
    PORTA.3=0;
    delay_ms(17);
    }
}

void r1 (void)
{
//koordinat (0,2)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
```

```
    PORTA.6=0;
    delay_us(400);
    PORTA.4=0;
    delay_us(100);
    PORTA.1=0;
    PORTA.2=0;
    PORTA.5=0;
    delay_us(270);
    PORTA.0=0;
    delay_us(230);
    PORTA.3=0;
    delay_ms(17);
    }
}

void s1 (void)
{
// koordinat (1,2)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(350);
    PORTA.4=0;
    delay_us(150);
    PORTA.1=0;
    PORTA.2=0;
    PORTA.5=0;
    delay_us(180);
    PORTA.0=0;
    delay_us(320);
    PORTA.3=0;
    delay_ms(17);
    }
}

void t1 (void)
{
//koordinat (2,2)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(300);
    PORTA.1=0;
    PORTA.2=0;
    PORTA.5=0;
    delay_us(100);
    PORTA.0=0;
    delay_us(350);
    PORTA.3=0;
    delay_ms(17);
    }
}

void u1 (void)
{
//koordinat (3,2)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(300);
    PORTA.1=0;
    PORTA.2=0;
    PORTA.5=0;
    PORTA.0=0;
    delay_us(450);
    PORTA.3=0;
    delay_ms(17);
    }
}

void v1 (void)
{
//koordinat (4,2)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(200);
    PORTA.0=0;
    delay_us(100);
    PORTA.1=0;
    PORTA.2=0;
    PORTA.5=0;
    delay_us(450);
    PORTA.3=0;
    delay_ms(17);
    }
}

void w1 (void)
{
//koordinat (5,2)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(80);
    PORTA.0=0;
    delay_us(220);
    PORTA.1=0;
    PORTA.2=0;
    PORTA.5=0;
    delay_us(450);
    PORTA.3=0;
    delay_ms(17);
    }
}

void x1 (void)
{
//koordinat (6,2)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.0=0;
```

```
    delay_us(300);
    PORTA.5=0;
    PORTA.1=0;
    PORTA.2=0;
    delay_us(50);
    PORTA.4=0;
    delay_us(550);
    PORTA.3=0;
    delay_ms(17);
    }
}

void y1 (void)
{
//koordinat (7,2)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(130);
    PORTA.0=0;
    delay_us(300);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(70);
    PORTA.5=0;
    delay_us(50);
    PORTA.4=0;
    delay_us(500);
    PORTA.3=0;
    delay_ms(17);
    }
}

void z1 (void)
{
//koordinat (0,3)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(300);
    PORTA.4=0;
    delay_us(50);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(150);
    PORTA.5=0;
    delay_us(220);
    PORTA.0=0;
    delay_us(30);
    PORTA.3=0;
    delay_ms(17);
    }
}

void a2 (void)
{
//koordinat (1,3)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(300);
    PORTA.4=0;
    delay_us(100);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(100);
    PORTA.5=0;
    delay_us(150);
    PORTA.0=0;
    delay_us(170);
    PORTA.3=0;
    delay_ms(17);
    }
}

void b2 (void)
{
//koordinat (2,3)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(250);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(50);
    PORTA.5=0;
    delay_us(80);
    PORTA.0=0;
    delay_us(200);
    PORTA.3=0;
    delay_ms(17);
    }
}

void c2 (void)
{
//koordinat (3,3)
    for(i=0;i<200;i++)
    {
     PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(250);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(50);
    PORTA.5=0;
    PORTA.0=0;
    delay_us(300);
    PORTA.3=0;
    delay_ms(17);
    }
}

void d2 (void)
{
//koordinat (4,3)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
```

```
    delay_us(200);
    PORTA.4=0;
    delay_us(200);
    PORTA.0=0;
    delay_us(50);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(50);
    PORTA.5=0;
    delay_us(300);
    PORTA.3=0;
    delay_ms(17);
    }
}

void e2 (void)
{
//koordinat (5,3)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(130);
    PORTA.0=0;
    delay_us(120);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(50);
    PORTA.5=0;
    delay_us(300);
    PORTA.3=0;
    delay_ms(17);
    }
}

void f2 (void)
{
//koordinat (6,3)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(250);
    PORTA.4=0;
    PORTA.0=0;
    delay_us(200);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(50);
    PORTA.5=0;
    delay_us(300);
    PORTA.3=0;
    delay_ms(17);
    }
}

void g2 (void)
{
//koordinat (7,3)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(150);
    PORTA.0=0;
    delay_us(100);
    PORTA.4=0;
    delay_us(100);
    PORTA.0=0;
    PORTA.1=0;
    PORTA.2=0;
    delay_us(150);
    PORTA.5=0;
    delay_us(200);
    PORTA.3=0;
    delay_ms(17);
    }
}

void h2 (void)
{
//koordinat (0,4)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(250);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(250);
    PORTA.5=0;
    PORTA.4=0;
    delay_us(200);
    PORTA.0=0;
    delay_us(50);
    PORTA.3=0;
    delay_ms(17);
    }
}

void j2 (void)
{
//koordinat (1,4)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(300);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(200);
    PORTA.5=0;
    PORTA.4=0;
    delay_us(130);
    PORTA.0=0;
    delay_us(200);
    PORTA.3=0;
    delay_ms(17);
    }
}

void k2 (void)
{
//koordinat (2,4)
    for(i=0;i<200;i++)
    {
    PORTA=255;
```

```
    delay_us(1000);
    PORTA.6=0;
    delay_us(300);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(200);
    PORTA.5=0;
    PORTA.4=0;
    delay_us(50);
    PORTA.0=0;
    delay_us(280);
    PORTA.3=0;
    delay_ms(17);
    }
}

void l2 (void)
{
//koordinat (3,4)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(330);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(180);
    PORTA.5=0;
    PORTA.4=0;
    PORTA.0=0;
    delay_us(330);
    PORTA.3=0;
    delay_ms(17);
    }
}

void m2 (void)
{
//koordinat (4,4)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(330);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(100);
    PORTA.0=0;
    delay_us(80);
    PORTA.5=0;
    PORTA.4=0;
    delay_us(330);
    PORTA.3=0;
    delay_ms(17);
    }
}

void n2 (void)
{
//koordinat (5,4)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(330);
    PORTA.1=0;
    PORTA.2=0;
    PORTA.0=0;
    delay_us(180);
    PORTA.5=0;
    PORTA.4=0;
    delay_us(330);
    PORTA.3=0;
    delay_ms(17);
    }
}

void o2 (void)
{
//koordinat (6,4)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(300);
    PORTA.0=0;
    delay_us(50);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(180);
    PORTA.5=0;
    PORTA.4=0;
    delay_us(330);
    PORTA.3=0;
    delay_ms(17);
    }
}

void p2 (void)
{
//koordinat (7,4)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.0=0;
    delay_us(50);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(280);
    PORTA.5=0;
    PORTA.4=0;
    delay_us(230);
    PORTA.3=0;
    delay_ms(17);
    }
}

void q2 (void)
{
// koordinat (0,5)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(150);
    PORTA.1=0;
```

```
    PORTA.2=0;
    delay_us(350);
    PORTA.5=0;
    delay_us(200);
    PORTA.0=0;
    PORTA.3=0;
    PORTA.4=0;
    delay_ms(17);
    }
}

void r2 (void)
{
// koordinat (1,5)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(300);
    PORTA.5=0;
    delay_us(100);
    PORTA.4=0;
    delay_us(50);
    PORTA.0=0;
    delay_us(50);
    PORTA.3=0;
    delay_ms(17);
    }
}

void s2 (void)
{
// koordinat (2,5)
    for(i=0;i<200;i++)
    {
     PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(300);
    PORTA.5=0;
    delay_us(80);
    PORTA.0=0;
    delay_us(20);
    PORTA.4=0;
    delay_us(100);
    PORTA.3=0;
    delay_ms(17);
    }
}

void t2 (void)
{
// koordinat (3,5)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(300);
    PORTA.5=0;
    PORTA.0=0;
    delay_us(50);
    PORTA.4=0;
    delay_us(150);
    PORTA.3=0;
    delay_ms(17);
    }
}

void u2 (void)
{
// koordinat (4,5)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(240);
    PORTA.0=0;
    delay_us(60);
    PORTA.5=0;
    delay_us(50);
    PORTA.4=0;
    delay_us(150);
    PORTA.3=0;
    delay_ms(17);
    }
 }

void v2 (void)
{
// koordinat (5,5)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(180);
    PORTA.0=0;
    delay_us(120);
    PORTA.5=0;
    delay_us(50);
    PORTA.4=0;
    delay_us(150);
    PORTA.3=0;
    delay_ms(17);
    }
}

void w2 (void)
{
// koordinat (6,5)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(200);
```

```
    PORTA.1=0;
    PORTA.2=0;
    delay_us(100);
    PORTA.0=0;
    delay_us(200);
    PORTA.5=0;
    delay_us(50);
    PORTA.4=0;
    delay_us(150);
    PORTA.3=0;
    delay_ms(17);
    }
}

void x2 (void)
{
// koordinat (7,5)
    for(i=0;i<200;i++)
    {
     PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(150);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(120);
    PORTA.0=0;
    delay_us(220);
    PORTA.5=0;
    delay_us(50);
    PORTA.4=0;
    delay_us(100);
    PORTA.3=0;
    delay_ms(17);
    }
}

void bn (void)
{
//koordinat (3,0)
    for(i=0;i<200;i++)
    {
    PORTA=255;
    delay_us(1000);
    PORTA.6=0;
    delay_us(500);
    PORTA.5=0;
    PORTA.0=0;
    delay_us(200);
    PORTA.4=0;
    delay_us(400);
    PORTA.1=0;
    PORTA.2=0;
    delay_us(200);
    PORTA.3=0;
    delay_ms(17);
    }
}

 void tp (void)
{
//perintah tutup
PORTA.6=1;
delay_us(1000);
PORTA.6=0;
delay_ms(19);
}

void uk (void)
{
//perintah buka
PORTA.6=1;
delay_us(1300);
PORTA.6=0;
delay_ms(19);
}

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

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

// Input/Output Ports initialization
// Port A initialization
// Func7=Out Func6=Out Func5=Out Func4=Out
Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0
State1=0 State0=0
PORTA=0x00;
DDRA=0xFF;

// 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=Out Func6=Out Func5=Out Func4=Out
Func3=Out Func2=Out Func1=Out Func0=Out
// State7=0 State6=0 State5=0 State4=0 State3=0 State2=0
State1=0 State0=0
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;

// USART initialization
// Communication Parameters: 8 Data, 1 Stop, No Parity
// USART Receiver: On
// USART Transmitter: On
// USART Mode: Asynchronous
// USART Baud rate: 9600
UCSRA=0x00;
UCSRB=0x98;
UCSRC=0x86;
UBRRH=0x00;
UBRRL=0x47;

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

// LCD module initialization
lcd_init(16);

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

while (1)
      {
      // Place your code here

//======track colour CMUCam2+=========//
      printf("om 0 3\r");
      delay_ms(100);
      printf("tc 80 180 16 30 16 30 \r");
      delay_ms(100);


if(koor[0]<=1 )//===============tidak ada data yang
dikirim
{
lcd_clear();
sprintf(ko,"apa lagi %d %d",koor[0],koor[1]);
lcd_puts(ko);
delay_ms(200);
}
else
//=======================================
================ada data yang dikirim
{

if(koor[1]>=1)//==============================
========================dua
      {
            if(koor[0]=97)
            {
                  switch (koor[1])
                  {

                  //depan A

                  case 97 :      lcd_clear();
                                 sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                                 lcd_puts(ko);
                                 delay_ms(300);
                                 for(i=0;i<=100;i++)aa();
                                 break;
                  case 98 :      lcd_clear();
                                 sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                                 lcd_puts(ko);
                                 delay_ms(300);
                                 for(i=0;i<=100;i++)ab();
                                 break;
                  case 99 :      lcd_clear();
                                 sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                                 lcd_puts(ko);
                                 delay_ms(300);
                                 for(i=0;i<=100;i++)ac();
                                 break;
                  case 100 :     lcd_clear();
                                 sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                                 lcd_puts(ko);
                                 delay_ms(300);
                                 for(i=0;i<=100;i++)ad();
                                 break;
                  case 101 :     lcd_clear();
                                 sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                                 lcd_puts(ko);
                                 delay_ms(300);
                                 for(i=0;i<=100;i++)ae();
                                 break;
                  case 102 :     lcd_clear();
                                 sprintf(ko,"2A %u
%u",koor[0],koor[1]);
```

```
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)af();
                break;
        case 103 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)ag();
                break;
        case 104 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)ah();
                break;
        case 105 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)ai();
                break;
        case 106 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)aj();
                break;
        case 107 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)ak();
                break;
        case 108 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)al();
                break;
        case 109 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)am();
                break;
        case 110 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)an();
                break;
        case 111 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)ao();
                break;
        case 112 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)ap();
                break;
        case 113 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)aq();
                break;
        case 114 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)ar();
                break;
        case 115 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)as();
                break;
        case 116 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)at();
                break;
        case 117 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)au();
                break;
        case 118 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)av();
                break;
        case 119 :      lcd_clear();
                sprintf(ko,"2A %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)aw();
                break;


            }
        mi=0;
        koor[0]=0;
        koor[1]=0;
        }
    }
    else
//=====================================
================satu
```

```
      {
      switch (koor[0])
      {
       case 33 :      lcd_clear();
                      sprintf(ko,"buka %u %u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)uk();
                      break;
       case 35 :      lcd_clear();
                      sprintf(ko,"tutup %u %u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)tp();
                      break;
       case 36 :      lcd_clear();
                      sprintf(ko,"siap %u %u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)bn();
                      break;
       case 97  :     lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)a();
       break;
       case 98  :     lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)b();
       break;
       case 99  :     lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)c();
       break;
       case 100  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)d();
       break;
       case 101  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)e();
       break;
       case 102  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)f();
       break;
       case 103  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)g();
       break;
       case 104  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)h();
       break;
       case 106  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)j();
       break;
       case 107  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)k();
       break;
       case 108  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)l();
       break;
       case 109  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)m();
       break;
       case 110  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)n();
       break;
       case 111  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)o();
       break;
       case 112  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)p();
       break;
       case 113  :    lcd_clear();
                      sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                      lcd_puts(ko);
                      delay_ms(300);
                      for(i=0;i<=100;i++)q();
       break;
       case 114  :    lcd_clear();
```

```
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)r();
        break;
        case 115 :      lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)s();
        break;
        case 116 :      lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)t();
        break;
        case 117 :      lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)u();
        break;
        case 118 :      lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)v();
        break;
        case 119 :      lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)w();
        break;
        case 120 :      lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)x();
        break;
        case 121 :      lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)y();
        break;
        case 122 :      lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)z();
        break;



        //DENGAN ANGKA 1 SAMPAI 3
        case 66 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)a1();
        break;
        case 67 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)b1();
        break;
        case 68 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)c1();
        break;
        case 69 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)d1();
        break;
        case 70 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)e1();
        break;
        case 71 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)f1();
        break;
        case 72 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)g1();
        break;
        case 73 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)h1();
        break;
        case 74 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)j1();
        break;
        case 75 :       lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
```

```
                for(i=0;i<=100;i++)k1();
        break;
        case 76 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)l1();
        break;
        case 77 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)m1();
        break;
        case 78 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)n1();
        break;
        case 79 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)o1();
        break;
        case 80 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)p1();
        break;
        case 81 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)q1();
        break;
        case 82 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)r1();
        break;
        case 83 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)s1();
        break;
        case 84 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)t1();
        break;
        case 37 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)u1();
        break;
        case 38 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)v1();
        break;
        case 39 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)w1();
        break;
        case 40 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)x1();
        break;
        case 41 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)y1();
        break;
        case 42 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)z1();
        break;
        case 43 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)a2();
        break;
        case 44 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)b2();
        break;
        case 45 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)c2();
        break;
        case 46 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
```

```
                for(i=0;i<=100;i++)d2();
        break;
        case 47 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)e2();
        break;
        case 48 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)f2();
        break;
        case 49 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)g2();
        break;
        case 50 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)h2();
        break;
        case 51 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)j2();
        break;
        case 52 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)k2();
        break;
        case 53 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)l2();
        break;
        case 54 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)m2();
        break;
        case 55 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)n2();
        break;
        case 56 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)o2();
        break;
        case 57 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)p2();
        break;
        case 58 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)q2();
        break;
        case 59 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)r2();
        break;
        case 60 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)s2();
        break;
        case 61 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)t2();
        break;
        case 62 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)u2();
        break;
        case 63 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)v2();
        break;
        case 64 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
                for(i=0;i<=100;i++)w2();
        break;
        case 65 :     lcd_clear();
                sprintf(ko,"koordinat %u
%u",koor[0],koor[1]);
                lcd_puts(ko);
                delay_ms(300);
```

```
            for(i=0;i<=100;i++)x2();
 break;


};
                                                    mi=0;
                                                    koor[0]=0;
                                                    }
                                                }
```

PROGRAM VISUAL BASIC

```
=======PROGRAM VISUAL BASIC PADA FORM
CODE=======

Private Sub Command1_Click()

Dim txt As String
Dim token As String

   txt = GetToken(Text2.Text, "ACK") & vbCrLf
   Do
      token = GetToken("", "ACK")
      If token = "" Then Exit Do
      txt = txt & token & vbCrLf
   Loop
RichTextBox1 = txt

End Sub

Private Sub Command10_Click()
MSComm5.Output = "$"
End Sub

Private Sub Command2_Click()
Dim a As String
a = RichTextBox1.Text

Text3.Text = Mid(a, 3, 2)
Text4.Text = Mid(a, 6, 4)

End Sub

Private Sub Command3_Click()
If Val(Text3.Text) >= 26 And Val(Text3.Text) <= 32
Then
  Text5.Text = 7
ElseIf Val(Text3.Text) >= 33 And Val(Text3.Text) <= 39
Then
   Text5.Text = 6
ElseIf Val(Text3.Text) >= 40 And Val(Text3.Text) <= 46
Then
   Text5.Text = 5
ElseIf Val(Text3.Text) >= 47 And Val(Text3.Text) <= 53
Then
   Text5.Text = 4
ElseIf Val(Text3.Text) >= 54 And Val(Text3.Text) <= 60
Then
   Text5.Text = 3
ElseIf Val(Text3.Text) >= 61 And Val(Text3.Text) <= 67
Then
   Text5.Text = 2
ElseIf Val(Text3.Text) >= 68 And Val(Text3.Text) <= 74
Then
   Text5.Text = 1
ElseIf Val(Text3.Text) >= 75 And Val(Text3.Text) <= 81
Then
   Text5.Text = 0
End If

If Val(Text4.Text) >= 12 And Val(Text4.Text) <= 27
Then
  Text6.Text = 7
ElseIf Val(Text4.Text) >= 28 And Val(Text4.Text) <= 42
Then
   Text6.Text = 6
ElseIf Val(Text4.Text) >= 43 And Val(Text4.Text) <= 57
Then
   Text6.Text = 5
ElseIf Val(Text4.Text) >= 58 And Val(Text4.Text) <= 72
Then
   Text6.Text = 4
ElseIf Val(Text4.Text) >= 73 And Val(Text4.Text) <= 87
Then
   Text6.Text = 3
ElseIf Val(Text4.Text) >= 88 And Val(Text4.Text) <=
102 Then
   Text6.Text = 2
ElseIf Val(Text4.Text) >= 103 And Val(Text4.Text) <=
117 Then
   Text6.Text = 1
ElseIf Val(Text4.Text) >= 118 And Val(Text4.Text) <=
132 Then
   Text6.Text = 0
End If
End Sub

Private Sub Command4_Click()
End
End Sub

Private Sub Command5_Click()
If CStr(Combo1.ItemData(Combo1.ListIndex)) = 1 Then
      MSComm5.Output = "a"
      Text8.Text = "0"
      Text9.Text = "0"
   ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) = 2
Then
      MSComm5.Output = "b"
       Text8.Text = "1"
      Text9.Text = "0"
   ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) = 3
Then
      MSComm5.Output = "c"
      Text8.Text = "2"
      Text9.Text = "0"
   ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) = 4
Then
      MSComm5.Output = "d"
      Text8.Text = "3"
      Text9.Text = "0"
   ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) = 5
Then
      MSComm5.Output = "e"
      Text8.Text = "4"
      Text9.Text = "0"
   ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) = 6
Then
      MSComm5.Output = "f"
      Text8.Text = "5"
      Text9.Text = "0"
   ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) = 7
Then
      MSComm5.Output = "g"
      Text8.Text = "6"
      Text9.Text = "0"
   ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) = 8
Then
      MSComm5.Output = "h"
      Text8.Text = "7"
```

```
        Text9.Text = "0"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) = 9
Then
        MSComm5.Output = "j"
        Text8.Text = "0"
        Text9.Text = "1"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
10 Then
        MSComm5.Output = "k"
        Text8.Text = "1"
        Text9.Text = "1"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
11 Then
        MSComm5.Output = "l"
        Text8.Text = "2"
        Text9.Text = "1"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
12 Then
        MSComm5.Output = "m"
        Text8.Text = "3"
        Text9.Text = "1"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
13 Then
        MSComm5.Output = "n"
        Text8.Text = "4"
        Text9.Text = "1"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
14 Then
        MSComm5.Output = "o"
        Text8.Text = "5"
        Text9.Text = "1"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
15 Then
        MSComm5.Output = "p"
        Text8.Text = "6"
        Text9.Text = "1"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
16 Then
        MSComm5.Output = "q"
        Text8.Text = "7"
        Text9.Text = "1"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
17 Then
        MSComm5.Output = "r"
        Text8.Text = "0"
        Text9.Text = "2"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
18 Then
        MSComm5.Output = "s"
        Text8.Text = "1"
        Text9.Text = "2"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
19 Then
        MSComm5.Output = "t"
        Text8.Text = "2"
        Text9.Text = "2"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
20 Then
        MSComm5.Output = "u"
        Text8.Text = "3"
        Text9.Text = "2"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
21 Then
        MSComm5.Output = "v"
        Text8.Text = "4"
        Text9.Text = "2"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
22 Then
        MSComm5.Output = "w"
        Text8.Text = "5"
        Text9.Text = "2"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
23 Then
        MSComm5.Output = "x"
        Text8.Text = "6"
        Text9.Text = "2"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
24 Then
        MSComm5.Output = "y"
        Text8.Text = "7"
        Text9.Text = "2"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
25 Then
        MSComm5.Output = "z"
        Text8.Text = "0"
        Text9.Text = "3"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
26 Then
        MSComm5.Output = "aa"
        Text8.Text = "1"
        Text9.Text = "3"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
27 Then
        MSComm5.Output = "ab"
        Text8.Text = "2"
        Text9.Text = "3"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
28 Then
        MSComm5.Output = "ac"
        Text8.Text = "3"
        Text9.Text = "3"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
29 Then
        MSComm5.Output = "ad"
        Text8.Text = "4"
        Text9.Text = "3"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
30 Then
        MSComm5.Output = "ae"
        Text8.Text = "5"
        Text9.Text = "3"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
31 Then
        MSComm5.Output = "af"
        Text8.Text = "6"
        Text9.Text = "3"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
32 Then
        MSComm5.Output = "ag"
        Text8.Text = "7"
        Text9.Text = "3"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
33 Then
        MSComm5.Output = "ah"
        Text8.Text = "0"
        Text9.Text = "4"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
34 Then
        MSComm5.Output = "ai"
        Text8.Text = "1"
        Text9.Text = "4"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
35 Then
```

```
        MSComm5.Output = "aj"
        Text8.Text = "2"
        Text9.Text = "4"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
36 Then
        MSComm5.Output = "ak"
        Text8.Text = "3"
        Text9.Text = "4"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
37 Then
        MSComm5.Output = "al"
        Text8.Text = "4"
        Text9.Text = "4"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
38 Then
        MSComm5.Output = "am"
        Text8.Text = "5"
        Text9.Text = "4"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
39 Then
        MSComm5.Output = "an"
        Text8.Text = "6"
        Text9.Text = "4"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
40 Then
        MSComm5.Output = "ao"
        Text8.Text = "7"
        Text9.Text = "4"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
41 Then
        MSComm5.Output = "ap"
        Text8.Text = "0"
        Text9.Text = "5"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
42 Then
        MSComm5.Output = "aq"
        Text8.Text = "1"
        Text9.Text = "5"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
43 Then
        MSComm5.Output = "ar"
        Text8.Text = "2"
        Text9.Text = "5"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
44 Then
        MSComm5.Output = "as"
        Text8.Text = "3"
        Text9.Text = "5"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
45 Then
        MSComm5.Output = "at"
        Text8.Text = "4"
        Text9.Text = "5"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
46 Then
        MSComm5.Output = "au"
        Text8.Text = "5"
        Text9.Text = "5"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
47 Then
        MSComm5.Output = "av"
        Text8.Text = "6"
        Text9.Text = "5"
    ElseIf CStr(Combo1.ItemData(Combo1.ListIndex)) =
48 Then
        MSComm5.Output = aw
        Text8.Text = "7"
        Text9.Text = "5"

    End If

End Sub

Private Sub Command6_Click()
If CStr(Combo2.ItemData(Combo2.ListIndex)) = 49 Then
        MSComm5.Output = "B"
        Text8.Text = "0"
        Text9.Text = "0"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
50 Then
        MSComm5.Output = "C"
        Text8.Text = "1"
        Text9.Text = "0"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
51 Then
        MSComm5.Output = "D"
        Text8.Text = "2"
        Text9.Text = "0"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
52 Then
        MSComm5.Output = "E"
        Text8.Text = "3"
        Text9.Text = "0"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
53 Then
        MSComm5.Output = "F"
        Text8.Text = "4"
        Text9.Text = "0"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
54 Then
        MSComm5.Output = "G"
        Text8.Text = "5"
        Text9.Text = "0"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
55 Then
        MSComm5.Output = "H"
        Text8.Text = "6"
        Text9.Text = "0"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
56 Then
        MSComm5.Output = "I"
        Text8.Text = "7"
        Text9.Text = "0"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
57 Then
        MSComm5.Output = "J"
        Text8.Text = "0"
        Text9.Text = "1"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
58 Then
        MSComm5.Output = "K"
        Text8.Text = "1"
        Text9.Text = "1"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
59 Then
        MSComm5.Output = "L"
        Text8.Text = "2"
        Text9.Text = "1"
    ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
60 Then
        MSComm5.Output = "M"
        Text8.Text = "3"
        Text9.Text = "1"
```

```
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
61 Then
      MSComm5.Output = "N"
      Text8.Text = "4"
      Text9.Text = "1"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
62 Then
      MSComm5.Output = "O"
      Text8.Text = "5"
      Text9.Text = "1"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
63 Then
      MSComm5.Output = "P"
      Text8.Text = "6"
      Text9.Text = "1"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
64 Then
      MSComm5.Output = "Q"
      Text8.Text = "7"
      Text9.Text = "1"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
65 Then
      MSComm5.Output = "R"
      Text8.Text = "0"
      Text9.Text = "2"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
66 Then
      MSComm5.Output = "S"
      Text8.Text = "1"
      Text9.Text = "2"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
67 Then
      MSComm5.Output = "T"
      Text8.Text = "2"
      Text9.Text = "2"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
68 Then
      MSComm5.Output = "%"
      Text8.Text = "3"
      Text9.Text = "2"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
69 Then
      MSComm5.Output = "&"
      Text8.Text = "4"
      Text9.Text = "2"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
70 Then
      MSComm5.Output = "'"
      Text8.Text = "5"
      Text9.Text = "2"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
71 Then
      MSComm5.Output = "("
      Text8.Text = "6"
      Text9.Text = "2"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
72 Then
      MSComm5.Output = ")"
      Text8.Text = "7"
      Text9.Text = "2"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
73 Then
      MSComm5.Output = "*"
      Text8.Text = "0"
      Text9.Text = "3"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
74 Then
      MSComm5.Output = "+"
      Text8.Text = "1"
      Text9.Text = "3"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
75 Then
      MSComm5.Output = ","
      Text8.Text = "2"
      Text9.Text = "3"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
76 Then
      MSComm5.Output = "-"
      Text8.Text = "3"
      Text9.Text = "3"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
77 Then
      MSComm5.Output = "."
      Text8.Text = "4"
      Text9.Text = "3"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
78 Then
      MSComm5.Output = "/"
      Text8.Text = "5"
      Text9.Text = "3"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
79 Then
      MSComm5.Output = "0"
      Text8.Text = "6"
      Text9.Text = "3"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
80 Then
      MSComm5.Output = "1"
      Text8.Text = "7"
      Text9.Text = "3"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
81 Then
      MSComm5.Output = "2"
      Text8.Text = "0"
      Text9.Text = "4"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
82 Then
      MSComm5.Output = "3"
      Text8.Text = "1"
      Text9.Text = "4"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
83 Then
      MSComm5.Output = "4"
      Text8.Text = "2"
      Text9.Text = "4"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
84 Then
      MSComm5.Output = "5"
      Text8.Text = "3"
      Text9.Text = "4"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
85 Then
      MSComm5.Output = "6"
      Text8.Text = "4"
      Text9.Text = "4"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
86 Then
      MSComm5.Output = "7"
      Text8.Text = "5"
      Text9.Text = "4"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
87 Then
      MSComm5.Output = "8"
      Text8.Text = "6"
```

```
      Text9.Text = "4"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
88 Then
      MSComm5.Output = "9"
      Text8.Text = "7"
      Text9.Text = "4"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
89 Then
      MSComm5.Output = ":"
      Text8.Text = "0"
      Text9.Text = "5"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
90 Then
      MSComm5.Output = ";"
      Text8.Text = "1"
      Text9.Text = "5"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
91 Then
      MSComm5.Output = "<"
      Text8.Text = "2"
      Text9.Text = "5"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
92 Then
      MSComm5.Output = "="
      Text8.Text = "3"
      Text9.Text = "5"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
93 Then
      MSComm5.Output = ">"
      Text8.Text = "4"
      Text9.Text = "5"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
94 Then
      MSComm5.Output = "?"
      Text8.Text = "5"
      Text9.Text = "5"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
95 Then
      MSComm5.Output = "@"
      Text8.Text = "6"
      Text9.Text = "5"
   ElseIf CStr(Combo2.ItemData(Combo2.ListIndex)) =
96 Then
      MSComm5.Output = "A"
      Text8.Text = "7"
      Text9.Text = "5"

   End If
End Sub

Private Sub Command7_Click()
      MSComm5.Output = "!"
End Sub

Private Sub Command8_Click()
      MSComm5.Output = "#"
End Sub

Private Sub Command9_Click()
If (Text5.Text <> "" And Text6.Text <> "") And
(Text8.Text <> "" And Text9.Text <> "") Then
   If (Text5.Text = Text8.Text) And (Text6.Text =
Text9.Text) Then
   Text7.Text = "benar"
   ElseIf (Text5.Text <> Text8.Text) And (Text6.Text =
Text9.Text) Then
   Text7.Text = "salah, x tidak sama"
   ElseIf (Text5.Text = Text8.Text) And (Text6.Text <>
Text9.Text) Then
   Text7.Text = "salah, y tidak sama"
   Else
   Text7.Text = "salah"
   End If

End If
End Sub

Private Sub Form_Load()
   MSComm5.PortOpen = True
   MSComm4.PortOpen = True

   'combo untuk posisi awal'
   Combo1.AddItem "posisi awal 0,0"
   Combo1.ItemData(Combo1.NewIndex) = 1
   Combo1.AddItem "posisi awal 1,0"
   Combo1.ItemData(Combo1.NewIndex) = 2
   Combo1.AddItem "posisi awal 2,0"
   Combo1.ItemData(Combo1.NewIndex) = 3
   Combo1.AddItem "posisi awal 3,0"
   Combo1.ItemData(Combo1.NewIndex) = 4
   Combo1.AddItem "posisi awal 4,0"
   Combo1.ItemData(Combo1.NewIndex) = 5
   Combo1.AddItem "posisi awal 5,0"
   Combo1.ItemData(Combo1.NewIndex) = 6
   Combo1.AddItem "posisi awal 6,0"
   Combo1.ItemData(Combo1.NewIndex) = 7
   Combo1.AddItem "posisi awal 7,0"
   Combo1.ItemData(Combo1.NewIndex) = 8
   Combo1.AddItem "posisi awal 0,1"
   Combo1.ItemData(Combo1.NewIndex) = 9
   Combo1.AddItem "posisi awal 1,1"
   Combo1.ItemData(Combo1.NewIndex) = 10
   Combo1.AddItem "posisi awal 2,1"
   Combo1.ItemData(Combo1.NewIndex) = 11
   Combo1.AddItem "posisi awal 3,1"
   Combo1.ItemData(Combo1.NewIndex) = 12
   Combo1.AddItem "posisi awal 4,1"
   Combo1.ItemData(Combo1.NewIndex) = 13
   Combo1.AddItem "posisi awal 5,1"
   Combo1.ItemData(Combo1.NewIndex) = 14
   Combo1.AddItem "posisi awal 6,1"
   Combo1.ItemData(Combo1.NewIndex) = 15
   Combo1.AddItem "posisi awal 7,1"
   Combo1.ItemData(Combo1.NewIndex) = 16
   Combo1.AddItem "posisi awal 0,2"
   Combo1.ItemData(Combo1.NewIndex) = 17
   Combo1.AddItem "posisi awal 1,2"
   Combo1.ItemData(Combo1.NewIndex) = 18
   Combo1.AddItem "posisi awal 2,2"
   Combo1.ItemData(Combo1.NewIndex) = 19
   Combo1.AddItem "posisi awal 3,2"
   Combo1.ItemData(Combo1.NewIndex) = 20
   Combo1.AddItem "posisi awal 4,2"
   Combo1.ItemData(Combo1.NewIndex) = 21
   Combo1.AddItem "posisi awal 5,2"
   Combo1.ItemData(Combo1.NewIndex) = 22
   Combo1.AddItem "posisi awal 6,2"
   Combo1.ItemData(Combo1.NewIndex) = 23
   Combo1.AddItem "posisi awal 7,2"
   Combo1.ItemData(Combo1.NewIndex) = 24
   Combo1.AddItem "posisi awal 0,3"
   Combo1.ItemData(Combo1.NewIndex) = 25
   Combo1.AddItem "posisi awal 1,3"
   Combo1.ItemData(Combo1.NewIndex) = 26
```

```
Combo1.AddItem "posisi awal 2,3"
Combo1.ItemData(Combo1.NewIndex) = 27
Combo1.AddItem "posisi awal 3,3"
Combo1.ItemData(Combo1.NewIndex) = 28
Combo1.AddItem "posisi awal 4,3"
Combo1.ItemData(Combo1.NewIndex) = 29
Combo1.AddItem "posisi awal 5,3"
Combo1.ItemData(Combo1.NewIndex) = 30
Combo1.AddItem "posisi awal 6,3"
Combo1.ItemData(Combo1.NewIndex) = 31
Combo1.AddItem "posisi awal 7,3"
Combo1.ItemData(Combo1.NewIndex) = 32
Combo1.AddItem "posisi awal 0,4"
Combo1.ItemData(Combo1.NewIndex) = 33
Combo1.AddItem "posisi awal 1,4"
Combo1.ItemData(Combo1.NewIndex) = 34
Combo1.AddItem "posisi awal 2,4"
Combo1.ItemData(Combo1.NewIndex) = 35
Combo1.AddItem "posisi awal 3,4"
Combo1.ItemData(Combo1.NewIndex) = 36
Combo1.AddItem "posisi awal 4,4"
Combo1.ItemData(Combo1.NewIndex) = 37
Combo1.AddItem "posisi awal 5,4"
Combo1.ItemData(Combo1.NewIndex) = 38
Combo1.AddItem "posisi awal 6,4"
Combo1.ItemData(Combo1.NewIndex) = 39
Combo1.AddItem "posisi awal 7,4"
Combo1.ItemData(Combo1.NewIndex) = 40
Combo1.AddItem "posisi awal 0,5"
Combo1.ItemData(Combo1.NewIndex) = 41
Combo1.AddItem "posisi awal 1,5"
Combo1.ItemData(Combo1.NewIndex) = 42
Combo1.AddItem "posisi awal 2,5"
Combo1.ItemData(Combo1.NewIndex) = 43
Combo1.AddItem "posisi awal 3,5"
Combo1.ItemData(Combo1.NewIndex) = 44
Combo1.AddItem "posisi awal 4,5"
Combo1.ItemData(Combo1.NewIndex) = 45
Combo1.AddItem "posisi awal 5,5"
Combo1.ItemData(Combo1.NewIndex) = 46
Combo1.AddItem "posisi awal 6,5"
Combo1.ItemData(Combo1.NewIndex) = 47
Combo1.AddItem "posisi awal 7,5"
Combo1.ItemData(Combo1.NewIndex) = 48

'combo untuk posisi tujuan'
Combo2.AddItem "posisi akhir 0,0"
Combo2.ItemData(Combo2.NewIndex) = 49
Combo2.AddItem "posisi akhir 1,0"
Combo2.ItemData(Combo2.NewIndex) = 50
Combo2.AddItem "posisi akhir 2,0"
Combo2.ItemData(Combo2.NewIndex) = 51
Combo2.AddItem "posisi akhir 3,0"
Combo2.ItemData(Combo2.NewIndex) = 52
Combo2.AddItem "posisi akhir 4,0"
Combo2.ItemData(Combo2.NewIndex) = 53
Combo2.AddItem "posisi akhir 5,0"
Combo2.ItemData(Combo2.NewIndex) = 54
Combo2.AddItem "posisi akhir 6,0"
Combo2.ItemData(Combo2.NewIndex) = 55
Combo2.AddItem "posisi akhir 7,0"
Combo2.ItemData(Combo2.NewIndex) = 56
Combo2.AddItem "posisi akhir 0,1"
Combo2.ItemData(Combo2.NewIndex) = 57
Combo2.AddItem "posisi akhir 1,1"
Combo2.ItemData(Combo2.NewIndex) = 58
Combo2.AddItem "posisi akhir 2,1"
Combo2.ItemData(Combo2.NewIndex) = 59
Combo2.AddItem "posisi akhir 3,1"
Combo2.ItemData(Combo2.NewIndex) = 60
Combo2.AddItem "posisi akhir 4,1"
Combo2.ItemData(Combo2.NewIndex) = 61
Combo2.AddItem "posisi akhir 5,1"
Combo2.ItemData(Combo2.NewIndex) = 62
Combo2.AddItem "posisi akhir 6,1"
Combo2.ItemData(Combo2.NewIndex) = 63
Combo2.AddItem "posisi akhir 7,1"
Combo2.ItemData(Combo2.NewIndex) = 64
Combo2.AddItem "posisi akhir 0,2"
Combo2.ItemData(Combo2.NewIndex) = 65
Combo2.AddItem "posisi akhir 1,2"
Combo2.ItemData(Combo2.NewIndex) = 66
Combo2.AddItem "posisi akhir 2,2"
Combo2.ItemData(Combo2.NewIndex) = 67
Combo2.AddItem "posisi akhir 3,2"
Combo2.ItemData(Combo2.NewIndex) = 68
Combo2.AddItem "posisi akhir 4,2"
Combo2.ItemData(Combo2.NewIndex) = 69
Combo2.AddItem "posisi akhir 5,2"
Combo2.ItemData(Combo2.NewIndex) = 70
Combo2.AddItem "posisi akhir 6,2"
Combo2.ItemData(Combo2.NewIndex) = 71
Combo2.AddItem "posisi akhir 7,2"
Combo2.ItemData(Combo2.NewIndex) = 72
Combo2.AddItem "posisi akhir 0,3"
Combo2.ItemData(Combo2.NewIndex) = 73
Combo2.AddItem "posisi akhir 1,3"
Combo2.ItemData(Combo2.NewIndex) = 74
Combo2.AddItem "posisi akhir 2,3"
Combo2.ItemData(Combo2.NewIndex) = 75
Combo2.AddItem "posisi akhir 3,3"
Combo2.ItemData(Combo2.NewIndex) = 76
Combo2.AddItem "posisi akhir 4,3"
Combo2.ItemData(Combo2.NewIndex) = 77
Combo2.AddItem "posisi akhir 5,3"
Combo2.ItemData(Combo2.NewIndex) = 78
Combo2.AddItem "posisi akhir 6,3"
Combo2.ItemData(Combo2.NewIndex) = 79
Combo2.AddItem "posisi akhir 7,3"
Combo2.ItemData(Combo2.NewIndex) = 80
Combo2.AddItem "posisi akhir 0,4"
Combo2.ItemData(Combo2.NewIndex) = 81
Combo2.AddItem "posisi akhir 1,4"
Combo2.ItemData(Combo2.NewIndex) = 82
Combo2.AddItem "posisi akhir 2,4"
Combo2.ItemData(Combo2.NewIndex) = 83
Combo2.AddItem "posisi akhir 3,4"
Combo2.ItemData(Combo2.NewIndex) = 84
Combo2.AddItem "posisi akhir 4,4"
Combo2.ItemData(Combo2.NewIndex) = 85
Combo2.AddItem "posisi akhir 5,4"
Combo2.ItemData(Combo2.NewIndex) = 86
Combo2.AddItem "posisi akhir 6,4"
Combo2.ItemData(Combo2.NewIndex) = 87
Combo2.AddItem "posisi akhir 7,4"
Combo2.ItemData(Combo2.NewIndex) = 88
Combo2.AddItem "posisi akhir 0,5"
Combo2.ItemData(Combo2.NewIndex) = 89
Combo2.AddItem "posisi akhir 1,5"
Combo2.ItemData(Combo2.NewIndex) = 90
Combo2.AddItem "posisi akhir 2,5"
Combo2.ItemData(Combo2.NewIndex) = 91
Combo2.AddItem "posisi akhir 3,5"
Combo2.ItemData(Combo2.NewIndex) = 92
```

```
    Combo2.AddItem "posisi akhir 4,3"
    Combo2.ItemData(Combo2.NewIndex) = 93
    Combo2.AddItem "posisi akhir 5,5"
    Combo2.ItemData(Combo2.NewIndex) = 94
    Combo2.AddItem "posisi akhir 6,5"
    Combo2.ItemData(Combo2.NewIndex) = 95
    Combo2.AddItem "posisi akhir 7,5"
    Combo2.ItemData(Combo2.NewIndex) = 96

End Sub

Private Sub RichTextBox1_Change()
Command2_Click
End Sub

Private Sub Text2_Change()
Command1_Click
End Sub

Private Sub Text3_Change()
Command3_Click
End Sub

Private Sub Text4_Change()
Command3_Click
End Sub

Private Sub Text8_Change()
Command9_Click
End Sub

Private Sub Text9_Change()
Command9_Click
End Sub

Private Sub Timer1_Timer()
    Text1.Text = MSComm5.Input
    Text2.Text = MSComm4.Input
End Sub

Private Sub Timer2_Timer()
MSComm4.Output = Text1.Text
End Sub
```

==========PROGRAM VISUAL BASIC PADA MODULE CODE==========

```
Public Function GetToken(new_txt As String, delimiter As _
    String) As String
Static txt As String
Dim pos As Integer

    ' Save new text.
    If new_txt <> "" Then txt = new_txt

    pos = InStr(txt, delimiter)
    If pos < 1 Then pos = Len(txt) + 1
    GetToken = Left$(txt, pos - 1)
    pos = Len(txt) - pos + 1 - Len(delimiter)
    If pos < 1 Then
        txt = ""
    Else
        txt = Right$(txt, pos)
    End If
End Function
```

LAMPIRAN - C

DATASHEET ATMEGA16

DATASHEET MAX232

Features

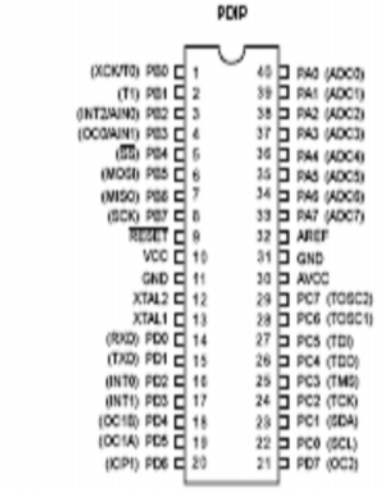
- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single-clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory segments
  - 16K Bytes of In-System Self-programmable Flash program memory
  - 512 Bytes EEPROM
  - 1K Byte Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C[1]
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - 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 QFN/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
- Power Consumption @ 1 MHz, 3V, and 25°C for ATmega16L
  - Active: 1.1 mA
  - Idle Mode: 0.35 mA
  - Power-down Mode: < 1 µA

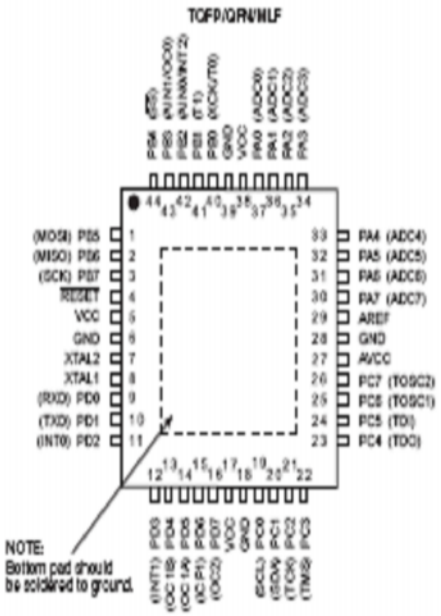
8-bit AVR® Microcontroller with 16K Bytes In-System Programmable Flash

ATmega16
ATmega16L

Pin Configurations

Figure 1. Pinout ATmega16





Disclaimer

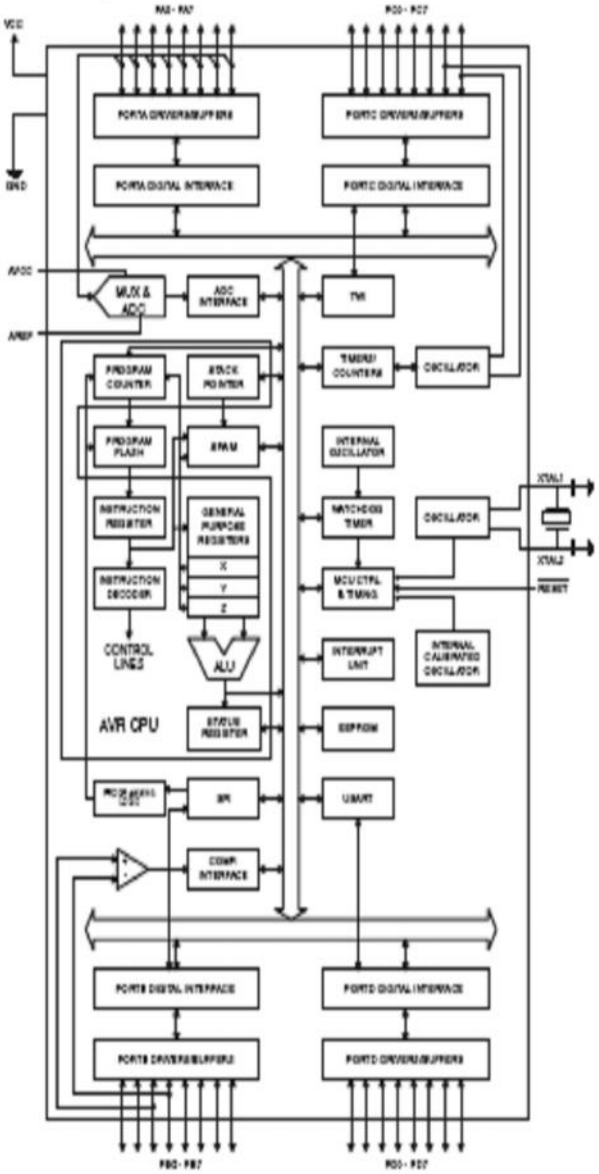
Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.

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.

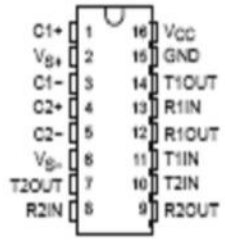
Port B (PB7..PB0)	Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if 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. Port B also serves the functions of various special features of the ATmega16 as listed on page 58.
Port C (PC7..PC0)	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. Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 61.
Port D (PD7..PD0)	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. Port D also serves the functions of various special features of the ATmega16 as listed on page 63.
$\overline{\text{RESET}}$	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 39. Shorter pulses are not guaranteed to generate a reset.
XTAL1	Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.
XTAL2	Output from the inverting Oscillator amplifier.
AVCC	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.
AREF	AREF is the analog reference pin for the A/D Converter.

MAX232, MAX232I
DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0-μF Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22
  - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1-μF Charge-Pump Capacitors is Available With the MAX202
- Applications
  - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
	SOP (NS)	Reel of 2000	MAX232NSR	MAX232
−40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

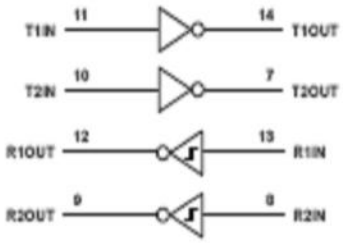
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



TEXAS INSTRUMENTS
POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, V_{CC} (see Note 1) ..... −0.3 V to 6 V
Positive output supply voltage range, V_{S+} ..... V_{CC} − 0.3 V to 15 V
Negative output supply voltage range, V_{S-} ..... −0.3 V to −15 V
Input voltage range, V_I: Driver ..... −0.3 V to V_{CC} + 0.3 V
Receiver ..... ±30 V
Output voltage range, V_O: T1OUT, T2OUT ..... V_{S-} − 0.3 V to V_{S+} + 0.3 V
R1OUT, R2OUT ..... −0.3 V to V_{CC} + 0.3 V
Short-circuit duration: T1OUT, T2OUT ..... Unlimited
Package thermal impedance, θ_{JA} (see Notes 2 and 3): D package ..... 73°C/W
DW package ..... 57°C/W
N package ..... 67°C/W
NS package ..... 64°C/W
Operating virtual junction temperature, T_J ..... 150°C
Storage temperature range, T_{stg} ..... −65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages are with respect to network GND.
2. Maximum power dissipation is a function of $T_J(max)$, θ_{JA}, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
3. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN,T2IN)		2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)				0.8	V
R1IN, R2IN	Receiver input voltage				±30	V
T_A	Operating free-air temperature	MAX232	0		70	°C
		MAX232I	−40		85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 4 and Figure 4)

PARAMETER		TEST CONDITIONS	MIN	TYP‡	MAX	UNIT
I_{CC}	Supply current	V_{CC} = 5.5 V, All outputs open, T_A = 25°C		8	10	mA

‡ All typical values are at V_{CC} = 5 V and T_A = 25°C.
NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	T1OUT, T2OUT	R_L = 3 kΩ to GND		5	7		V
V_{OL}	Low-level output voltage‡	T1OUT, T2OUT	R_L = 3 kΩ to GND			–7	–5	V
r_o	Output resistance	T1OUT, T2OUT	V_{S+} = V_{S-} = 0,	V_O = ±2 V	300			Ω
I_{OS}§	Short-circuit output current	T1OUT, T2OUT	V_{CC} = 5.5 V,	V_O = 0		±10		mA
I_{IS}	Short-circuit input current	T1IN, T2IN	V_I = 0				200	µA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
§ Not more than one output should be shorted at a time.
NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R_L = 3 kΩ to 7 kΩ, See Figure 2			30	V/µs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/µs
	Data rate	One TOUT switching		120		kbit/s

NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	R1OUT, R2OUT	I_{OH} = –1 mA		3.5			V
V_{OL}	Low-level output voltage‡	R1OUT, R2OUT	I_{OL} = 3.2 mA				0.4	V
V_{IT+}	Receiver positive-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V,	T_A = 25°C		1.7	2.4	V
V_{IT-}	Receiver negative-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V,	T_A = 25°C	0.8	1.2		V
V_{hys}	Input hysteresis voltage	R1IN, R2IN	V_{CC} = 5 V		0.2	0.5	1	V
r_i	Receiver input resistance	R1IN, R2IN	V_{CC} = 5,	T_A = 25°C	3	5	7	kΩ

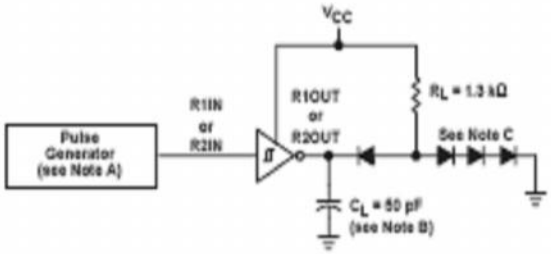
† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4 and Figure 1)

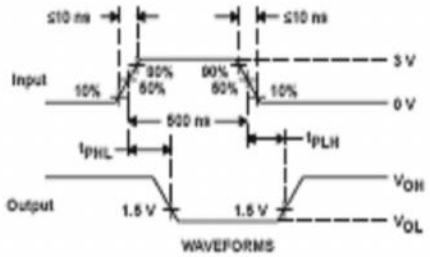
PARAMETER		TYP	UNIT
$t_{PLH(R)}$	Receiver propagation delay time, low- to high-level output	500	ns
$t_{PHL(R)}$	Receiver propagation delay time, high- to low-level output	500	ns

NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT

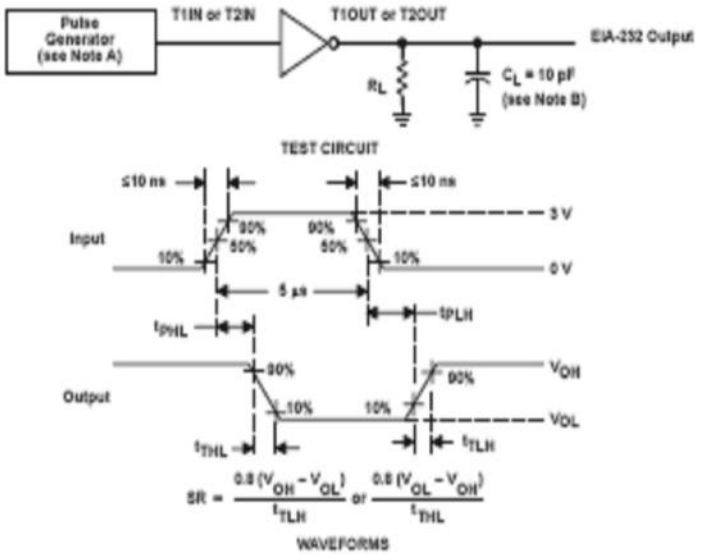


WAVEFORMS

NOTES: A. The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.
B. C_L includes probe and jig capacitance.
C. All diodes are 1N3064 or equivalent.

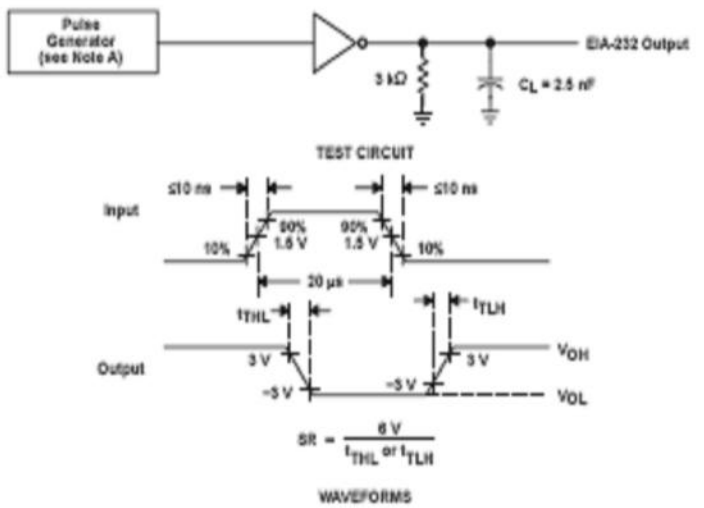
Figure 1. Receiver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements

PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.
B. C_L includes probe and jig capacitance.

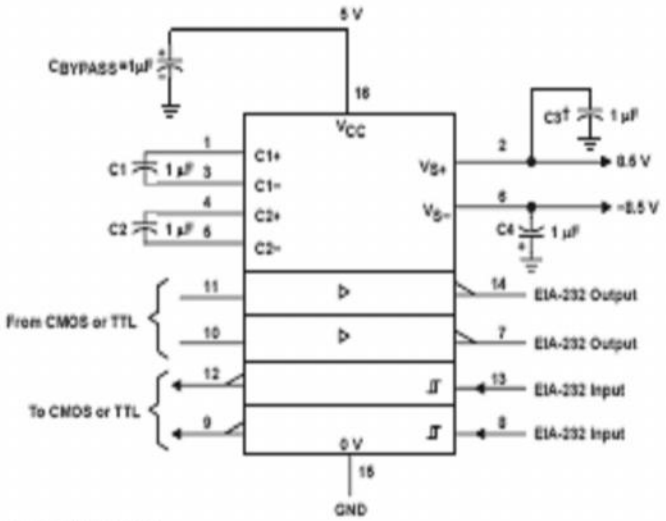
Figure 2. Driver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements (5-μs Input)



NOTE A: The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.

Figure 3. Test Circuit and Waveforms for t_{THL} and t_{TLH} Measurements (20-μs Input)

APPLICATION INFORMATION



†C3 can be connected to V_{CC} or GND.

NOTES: A. Resistor values shown are nominal.

B. Nonpolarized ceramic capacitors are acceptable. If polarized tantalum or electrolytic capacitors are used, they should be connected as shown. In addition to the 1-µF capacitors shown, the MAX202 can operate with 0.1-µF capacitors.

Figure 4. Typical Operating Circuit

PACKAGING INFORMATION

Orderable Device	Status [1]	Package Type	Package Drawing	Pins	Package Qty	Eco Plan [2]	Lead/Ball Finish	MSL Peak Temp [3]
MAX232D	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DE4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DG4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DR	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DRE4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DRG4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DW	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWE4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWG4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWR	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWRE4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232DWRG4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232I	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDE4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDG4	ACTIVE	SOIC	D	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDR	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDRE4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDRG4	ACTIVE	SOIC	D	16	2500	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDW	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWE4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWG4	ACTIVE	SOIC	DW	16	40	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWR	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWRE4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IDWRG4	ACTIVE	SOIC	DW	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232IN	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type

TEXAS INSTRUMENTS
www.ti.com

PACKAGE OPTION ADDENDUM

4-Jun-2007

Orderable Device	Status [1]	Package Type	Package Drawing	Pins	Package Qty	Eco Plan [2]	Lead/Ball Finish	MSL Peak Temp [3]
MAX232INE4	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232N	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232NE4	ACTIVE	PDIP	N	16	25	Pb-Free (RoHS)	CU NIPDAU	N / A for Pkg Type
MAX232NSR	ACTIVE	SO	NS	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232NSRE4	ACTIVE	SO	NS	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM
MAX232NSRG4	ACTIVE	SO	NS	16	2000	Green (RoHS & no Sb/Br)	CU NIPDAU	Level-1-260C-UNLIM

[1] The marketing status values are defined as follows:
ACTIVE: Product device recommended for new designs.
LIFEBUY: TI has announced that the device will be discontinued, and a lifetime-buy period is in effect.
NRND: Not recommended for new designs. Device is in production to support existing customers, but TI does not recommend using this part in a new design.
PREVIEW: Device has been announced but is not in production. Samples may or may not be available.
OBSOLETE: TI has discontinued the production of the device.

[2] Eco Plan - The planned eco-friendly classification: Pb-Free (RoHS), Pb-Free (RoHS Exempt), or Green (RoHS & no Sb/Br) - please check http://www.ti.com/productcontent for the latest availability information and additional product content details.
TBD: The Pb-Free/Green conversion plan has not been defined.
Pb-Free (RoHS): TI's terms "Lead-Free" or "Pb-Free" mean semiconductor products that are compatible with the current RoHS requirements for all 6 substances, including the requirement that lead not exceed 0.1% by weight in homogeneous materials. Where designed to be soldered at high temperatures, TI Pb-Free products are suitable for use in specified lead-free processes.
Pb-Free (RoHS Exempt): This component has a RoHS exemption for either 1) lead-based flip-chip solder bumps used between the die and package, or 2) lead-based die adhesive used between the die and leadframe. The component is otherwise considered Pb-Free (RoHS compatible) as defined above.
Green (RoHS & no Sb/Br): TI defines "Green" to mean Pb-Free (RoHS compatible), and free of Bromine (Br) and Antimony (Sb) based flame retardants (Br or Sb do not exceed 0.1% by weight in homogeneous material)

[3] MSL, Peak Temp. -- The Moisture Sensitivity Level rating according to the JEDEC industry standard classifications, and peak solder temperature.

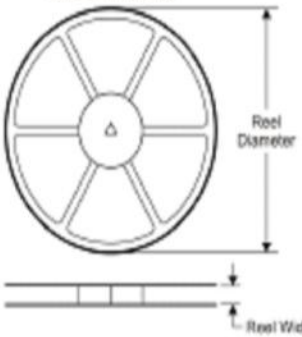
TEXAS INSTRUMENTS
www.ti.com

PACKAGE MATERIALS INFORMATION

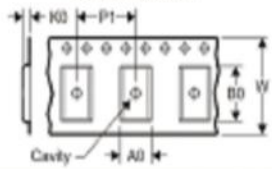
19-Mar-2008

TAPE AND REEL INFORMATION

REEL DIMENSIONS

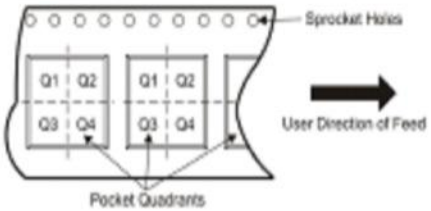


TAPE DIMENSIONS



A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

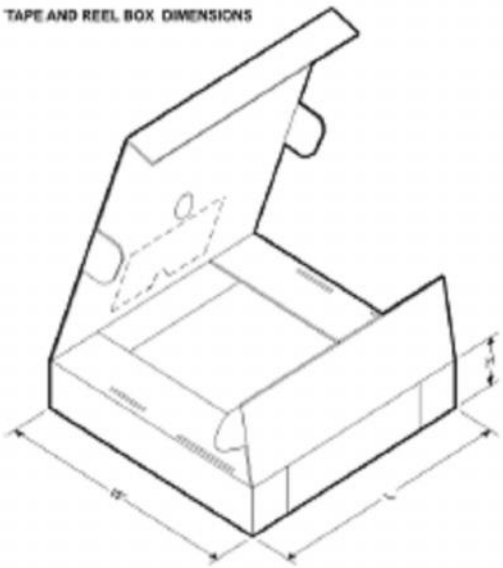
QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Reel Diameter (mm)	Reel Width W1 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
MAX232DR	SOIC	D	16	2500	330.0	16.4	6.5	10.3	2.1	8.0	16.0	Q1
MAX232DR	SOIC	D	16	2500	330.0	16.4	6.5	10.3	2.1	8.0	16.0	Q1
MAX232DWR	SOIC	DW	16	2000	330.0	16.4	10.75	10.7	2.7	12.0	16.0	Q1
MAX232IDR	SOIC	D	16	2500	330.0	16.4	6.5	10.3	2.1	8.0	16.0	Q1
MAX232IDWR	SOIC	DW	16	2000	330.0	16.4	10.75	10.7	2.7	12.0	16.0	Q1
MAX232NSR	SO	NS	16	2000	330.0	16.4	8.2	10.5	2.5	12.0	16.0	Q1

TAPE AND REEL BOX DIMENSIONS



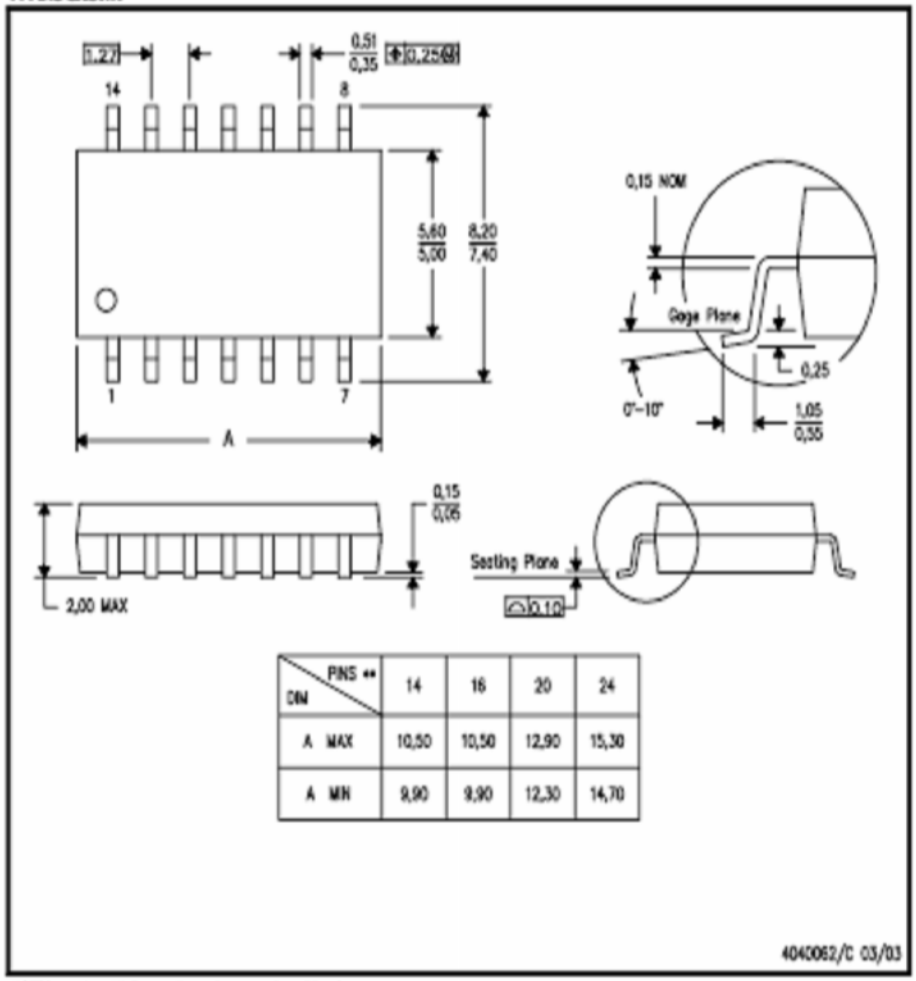
*All dimensions are nominal

Device	Package Type	Package Drawing	Pins	SPQ	Length (mm)	Width (mm)	Height (mm)
MAX232DR	SOIC	D	16	2500	346.0	346.0	33.0
MAX232DR	SOIC	D	16	2500	333.2	345.9	28.6
MAX232DWR	SOIC	DW	16	2000	346.0	346.0	33.0
MAX232IDR	SOIC	D	16	2500	333.2	345.9	28.6
MAX232IDWR	SOIC	DW	16	2000	346.0	346.0	33.0
MAX232NSR	SO	NS	16	2000	346.0	346.0	33.0

MECHANICAL DATA

NS (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

14-PINS SHOWN

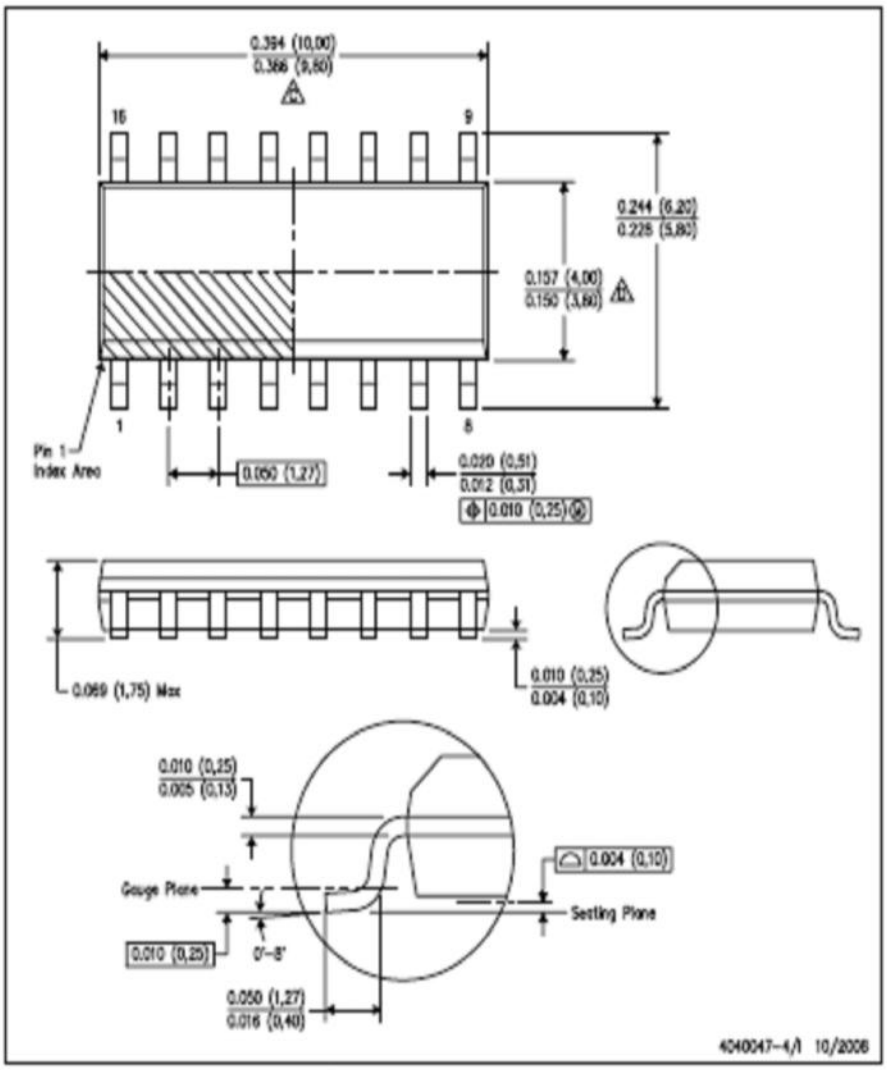


DIM \ PINS **	14	16	20	24
A MAX	10,50	10,50	12,90	15,30
A MIN	9,90	9,90	12,30	14,70

4040062/C 03/03

NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0,15.

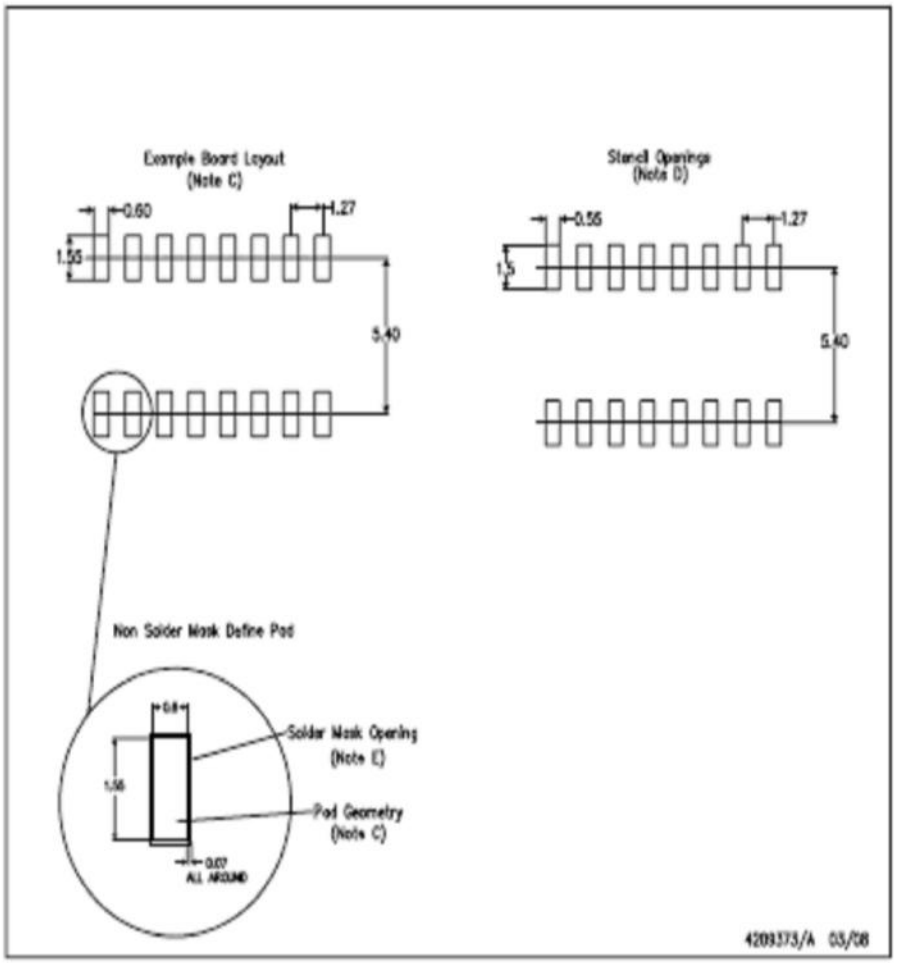
D (R–PDSO–G16) PLASTIC SMALL–OUTLINE PACKAGE



NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body length does not include mold flash, protrusions, or gate burrs. Mold flash, protrusions, or gate burrs shall not exceed .006 (0,15) per end.
D. Body width does not include interlead flash. Interlead flash shall not exceed .017 (0,43) per side.
E. Reference JEDEC MS–012 variation AC.

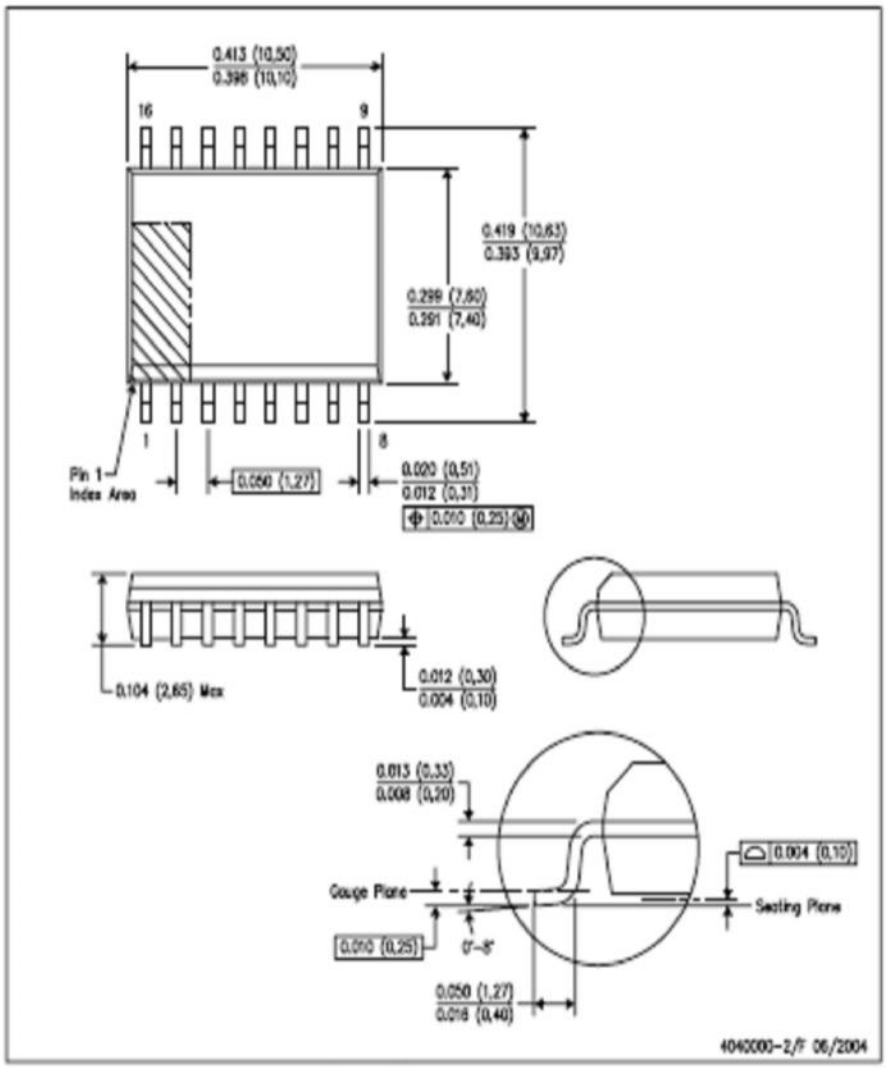
LAND PATTERN

D(R-PDSO-G16)



NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Refer to IPC7351 for alternate board design.
D. Laser cutting apertures with trapezoidal walls and also rounding corners will offer better paste release. Customers should contact their board assembly site for stencil design recommendations. Refer to IPC-7525
E. Customers should contact their board fabrication site for solder mask tolerances between and around signal pads.

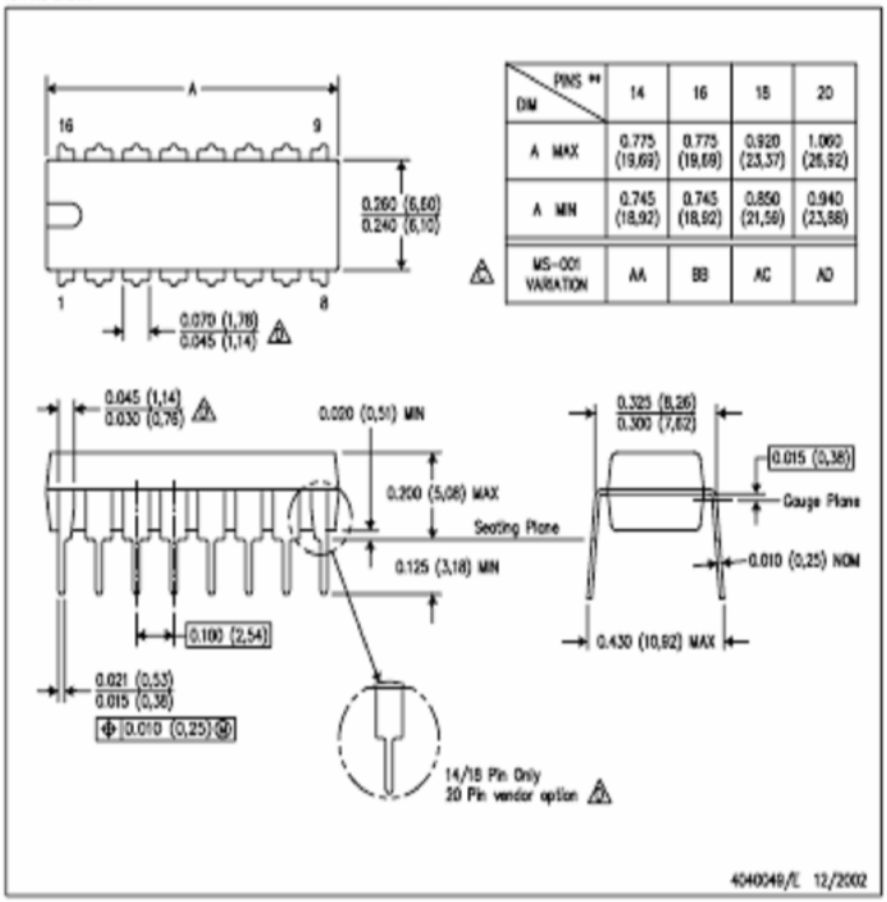
DW (R–PDSO–G16) PLASTIC SMALL–OUTLINE PACKAGE



NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0,15).
D. Falls within JEDEC MS–013 variation AA.

N (R-PDIP-T**) PLASTIC DUAL-IN-LINE PACKAGE

16 PINS SHOWN



DIM \ PINS **	14	16	18	20
A MAX	0.775 (19,69)	0.775 (19,69)	0.920 (23,37)	1.060 (26,92)
A MIN	0.745 (18,92)	0.745 (18,92)	0.850 (21,59)	0.940 (23,88)
MS-001 VARIATION	AA	BB	AC	AD

NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS-001, except 18 and 20 pin minimum body length (Dim A).
D. The 20 pin end lead shoulder width is a vendor option, either half or full width.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Clocks and Timers	www.ti.com/clocks	Digital Control	www.ti.com/digitalcontrol
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
RF/IF and ZigBee® Solutions	www.ti.com/lprf	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265