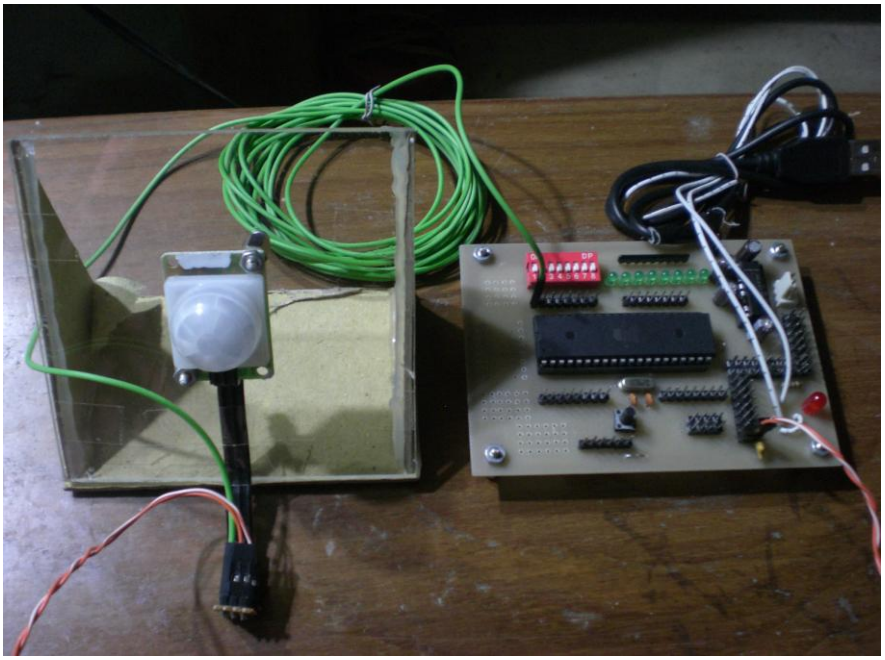
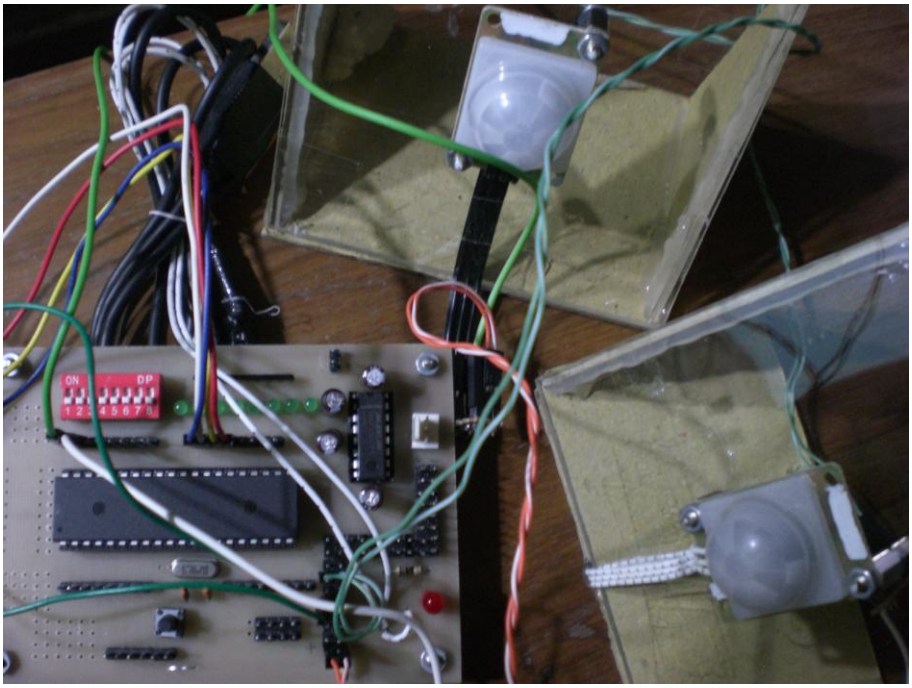
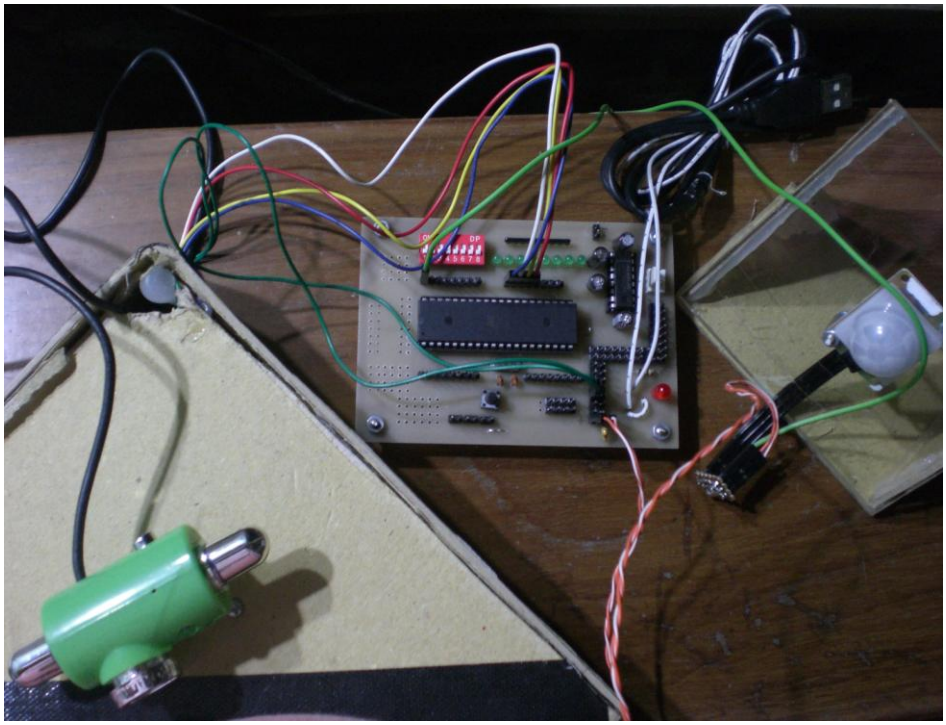
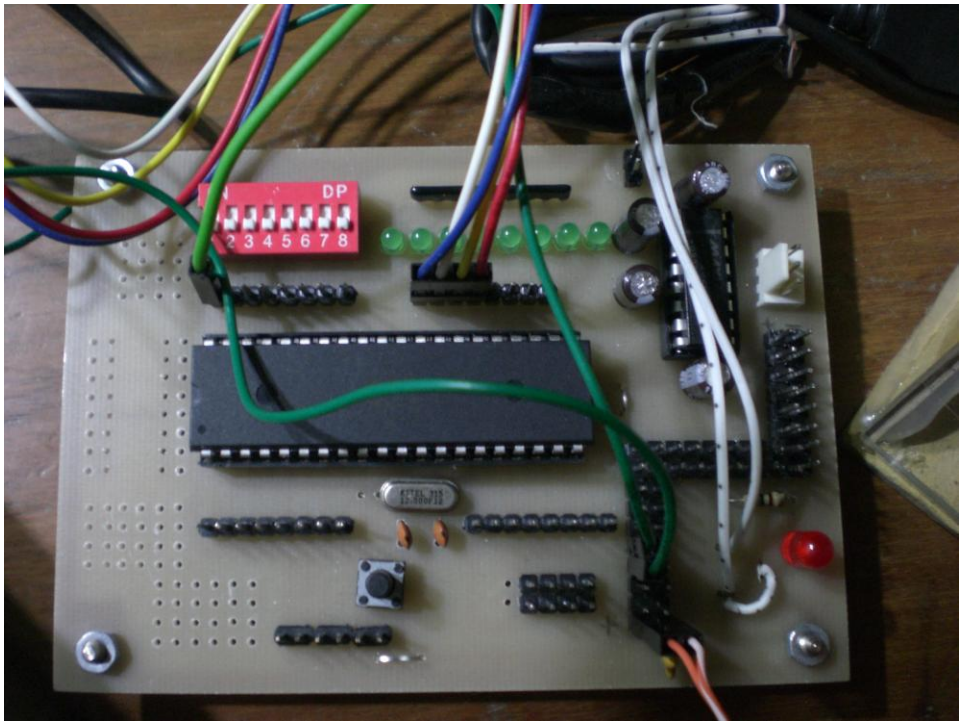
LAMPIRAN A

FOTO ALAT








DP
1 2 3 4 5 6 7 8


Sensor *PIR* 2
Sensor *PIR* 1
Webcam
AVR ATMega 16

LAMPIRAN B

PROGRAM CODEVISION

```
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 :
Version :
Date    : 6/3/2010
Author  : F4CG
Company : F4CG
Comments:
Chip type           : ATmega16
Program type        : Application
Clock frequency     : 12.000000 MHz
Memory model        : Small
External SRAM size  : 0
Data Stack size     : 256
*****************************************************/
#include <mega16.h>
#include <delay.h>
// Standard Input/Output functions
#include <stdio.h>
#define ADC_VREF_TYPE 0x00
// Read the AD conversion result
unsigned int read_adc(unsigned char adc_input)
{
ADMUX=adc_input | (ADC_VREF_TYPE & 0xff);
// Start the AD conversion
ADCSRA|=0x40;
// Wait for the AD conversion to complete
while ((ADCSRA & 0x10)==0);
ADCSRA|=0x10;
return ADCW;
}
// Declare your global variables here
unsigned int PIR_1;
unsigned int PIR_2;
float dataPIR_1;
float dataPIR_2;
int temp1;
int temp2;
```

```
int a=0,b=0,c=0;
unsigned int cekPir1()
{
PIR_1 = read_adc(0);
delay_ms(50);
dataPIR_1 = (float)PIR_1 * 0.0488;
if(dataPIR_1 > 20) temp1 = 1;
else temp1 = 0;
return temp1;
}
unsigned int cekPir2()
{
PIR_2 = read_adc(1);
delay_ms(50);
dataPIR_2 = (float)PIR_2 * 0.0488;
if(dataPIR_2 > 20) temp2 = 1;
else temp2 = 0;
return temp2;
}
void main(void)
{
int cek1;
int cek2;
// Input/Output Ports initialization
// Port A initialization
PORTA=0x00;
DDRA=0x00;
// Port B initialization
PORTB=0x00;
DDRB=0x00;
// Port C initialization
PORTC=0xff;
DDRC=0xff;
// Port D initialization
PORTD=0x00;
DDRD=0x00;
// 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=0x18;
UCSRC=0x86;
```

```
UBRRH=0x00;
UBRRL=0x4D;
// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
ACSR=0x80;
SFIOR=0x00;
// ADC initialization
// ADC Clock frequency: 187.500 kHz
// ADC Voltage Reference: AREF pin
// ADC Auto Trigger Source: None
ADMUX=ADC_VREF_TYPE & 0xff;
ADCSRA=0x86;
while (1)
      {
      cek1 = cekPir1();
      if(cek1 == 1)
      {
        PORTB=0xfe;
        while(b<=3)
        {
        printf("9");
        PORTC = 0b11111110;
        delay_ms(50);
        PORTC = 0b11111101;
        delay_ms(50);
        PORTC = 0b11111011;
        delay_ms(50);
        PORTC = 0b11110111;
        delay_ms(50);
        b++;
        }
      }
      cek2 = cekPir2();
      if(cek2 == 1)
      {
        PORTB=0xfe;
        while(a<=3)
        {
        printf("9");
        PORTC = 0b11110111;
        delay_ms(50);
        PORTC = 0b11111011;
        delay_ms(50);
        PORTC = 0b11111101;
        delay_ms(50);
```

```
      PORTC = 0b11111110;
      delay_ms(50);
      a++;
      }
    }
    delay_ms(50);
    a=0;
    b=0;
    c=0;
    };
}
```

LAMPIRAN C

PROGRAM BORLAND DELPHI

```
unit unVidRoll;

interface

uses

  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, JPEG, VideoCap;
type
  TForm1 = class(TForm)
    imVideoLabel: TLabel;
    pnVideoArea: TPanel;
    Image1: TImage;
    Timer1: TTimer;
    btCapture: TButton;
    procedure Timer1Timer(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure btCaptureClick(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;


var
  Form1: TForm1;
  buffer : byte;
```

```
implementation

function Out32(wAddr:word;bOut:byte):byte; stdcall; external 'inpout32.dll';
function Inp32(wAddr:word):byte; stdcall; external 'inpout32.dll';

{$R *.dfm}

procedure InitCapDevice(VideoArea : TWinControl; VideoLabel : TLabel);
Var MyCapStatusProc : TCapStatusProc;
Begin
  CapSetVideoArea( VideoArea );
  CapSetInfoLabel( VideoLabel );
  MyCapStatusProc := Nil;
  CapSetStatusProcedure( MyCapStatusProc );
  if CapOpenDriver then CapShow;
End;

Procedure CaptureImage( vimName : String; ImageArea : TImage);
Var vSaveJPEG : TJPEGImage;
Begin
  ImageArea.Picture := nil;
  CapSetSingleImageFileName( 'temp.BMP' );
  CapGrabSingleFrame;
  CapSetVideoLive;
  ImageArea.Picture.LoadFromFile( 'temp.BMP');
  vSaveJPEG := TJPEGImage.Create;
  vSaveJPEG.CompressionQuality := 40; {Compres : Quality}
  vSaveJPEG.Assign( ImageArea.Picture.Bitmap );
  vSaveJPEG.SaveToFile( 'foto\' + vimName + '.JPG');
  vSaveJPEG.Free;
End;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);
begin
buffer := inp32($3F8);
if buffer = $39 then
begin
btcapture.Click;
end;
end;

procedure TForm1.FormShow(Sender: TObject);
begin
InitCapDevice(pnVideoArea, imVideoLabel );
end;

procedure TForm1.btCaptureClick(Sender: TObject);
begin
CaptureImage(Formatdatetime('mmddhhnnss', now), Image1);
end;
end.
```

LAMPIRAN D

SKEMATIK RANGKAIAN