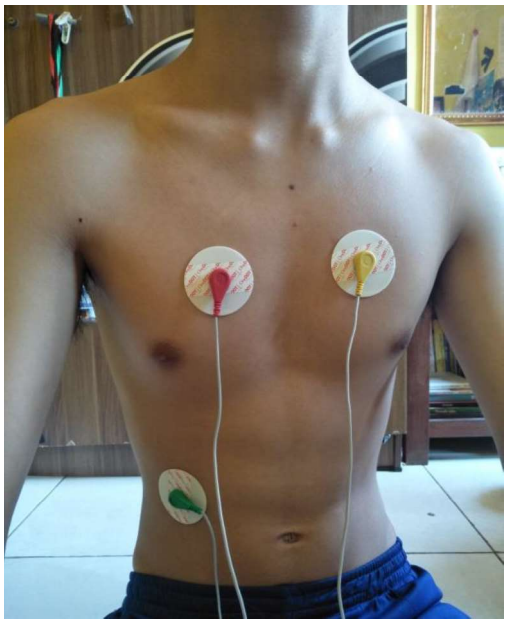
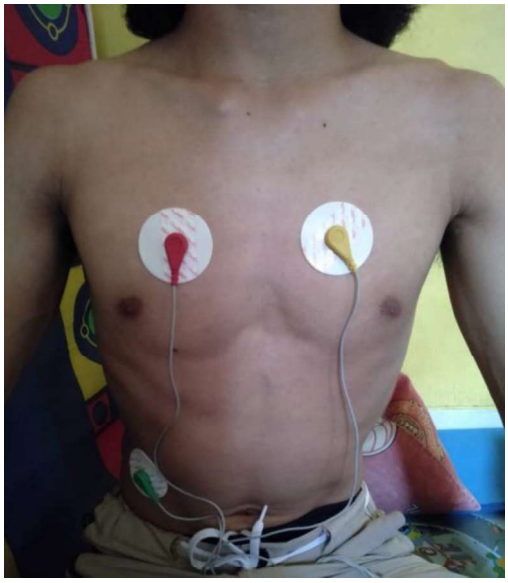
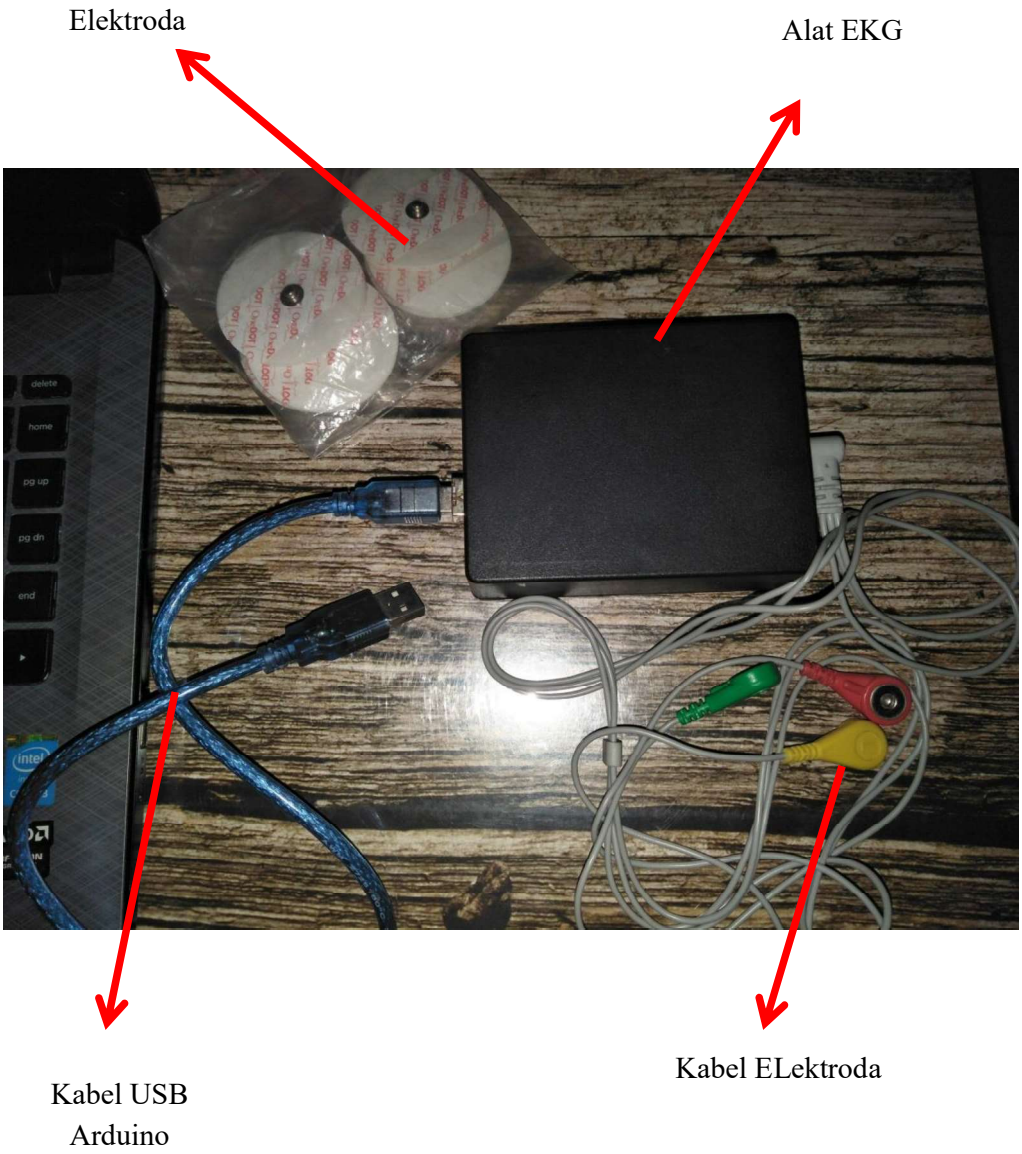
LAMPIRAN

Lampiran 1. Foto Pemasangan Elektroda Pada Responden



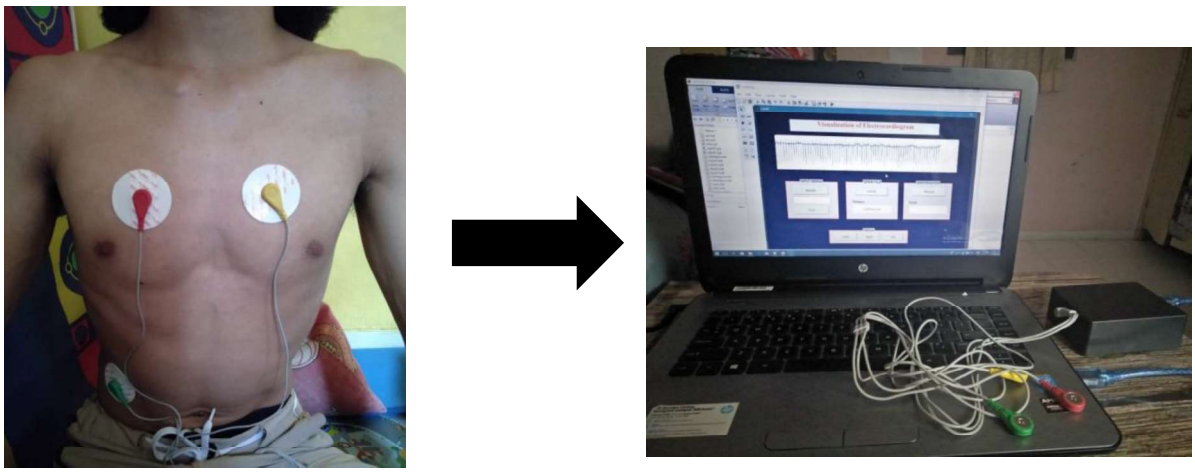


Lampiran 2. Gambar Alat

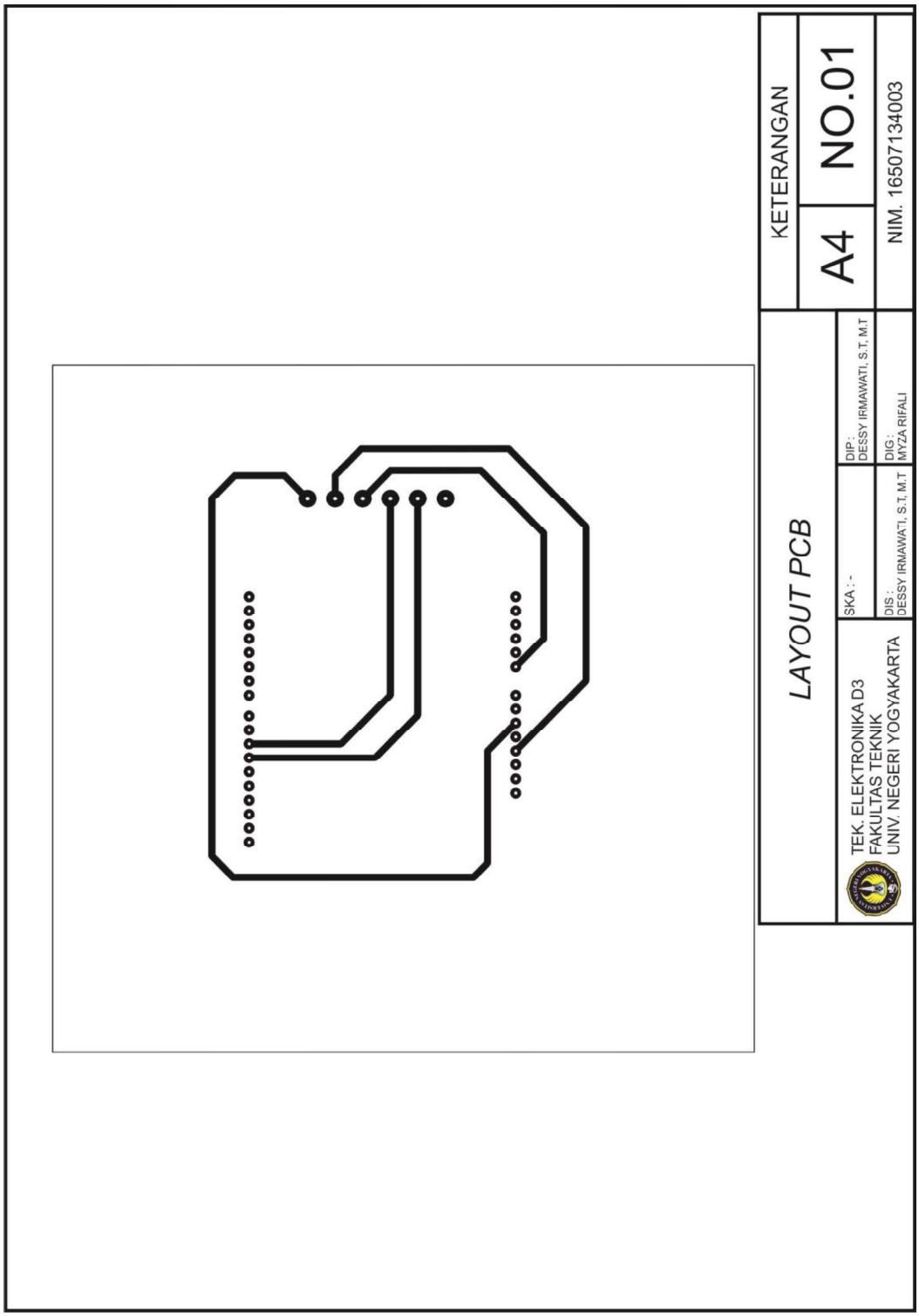


Lampiran 3. Petunjuk Penggunaan

1. Menghubungkan sumber 3,3V ke *input* tegangan
2. Memasang elektroda pada titik di tubuh responden yang telah ditentukan dengan posisi duduk tegak
3. Membuka aplikasi sistem cerdas pada dekstop Laptop/PC
4. Kemudian klik tombol rekam untuk melakukan perekaman
5. Hasil rekaman dapat disimpan dengan memilih tombol simpan *file*
6. Selanjutnya klik tombol pilih *file* dan pilih yang telah disimpan atau *file* yang ingin dilihat hasil klasifikasinya
7. Kemudian tekan tombol proses
8. Lihat pada kolom *output* maka secara otomatis hasil dari klasifikasi akan terlihat diikuti juga dengan munculnya grafik sinyal EKG.



Lampiran 4.Layout PCB



Lampiran 5. Source Code Arduino

```
void setup() {
  pinMode(10,INPUT);
  pinMode(11,INPUT);


}


void loop() {
  Serial.begin(9600);
  if((digitalRead(10)==1)||(digitalRead(11)==1)){
     Serial.println("Gagal");
  }
  else{
     Serial.println(analogRead(A0));
  }
```

Lampiran 6. Source Code Pelatihan dengan JST

```
clc;
clear;
close all;
warning off all;

data_folder = 'DATA LATIH';
filenames = dir(fullfile(data_folder, '*.mat'));
total_data = numel(filenames);

for n = 1:total_data;
full_name= fullfile(data_folder, filenames(n).name);
% Ekstraksi Ciri Data Input
CiriR(n) = mean(full_name);
CiriVAR(n) = var(full_name);
CiriSKEW(n) = skewness(full_name);
CiriKURT(n) = kurtosis(full_name);
CiriSD(n)= std(full_name);
end

x=[CiriR;CiriVAR;CiriSKEW;CiriKURT;CiriSD];
% Pembentukan target latih
target=zeros(1,82);
target(:,1:41)=1;
target(:,42:82)=2;



net = newff(x,target,[10 5],{'logsig','logsig'},'trainlm');
net.trainParam.epochs = 1000;
net.trainParam.goal = 1e-6;
net = train(net,x,target);
output = round(sim(net,x));
save net3.mat net

% Hasil identifikasi
%hasil_latih = round(sim(hasilnet,input));

[m,n] = find(output==target);
akurasi = sum(m)/total_data*100
```

Lampiran 7. Source Code GUI Matlab

```
function varargout = coba6(varargin)
% COBA6 MATLAB code for coba6.fig
%      COBA6, by itself, creates a new COBA6 or raises the existing
%      singleton*.
%
%      H = COBA12 returns the handle to a new COBA6 or the handle to
%      the existing singleton*.
%
%      COBA6('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in COBA6.M with the given input arguments.
%
%      COBA6('Property','Value',...) creates a new COBA6 or raises the
%      existing singleton*.  Starting from the left, property value pairs
are
%      applied to the GUI before coba6_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to coba6_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help coba6

% Last Modified by GUIDE v2.5 08-Jul-2019 10:21:51

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @coba6_OpeningFcn, ...
                   'gui_OutputFcn',  @coba6_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT



% --- Executes just before coba6 is made visible.
function coba6_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to coba6 (see VARARGIN)
```

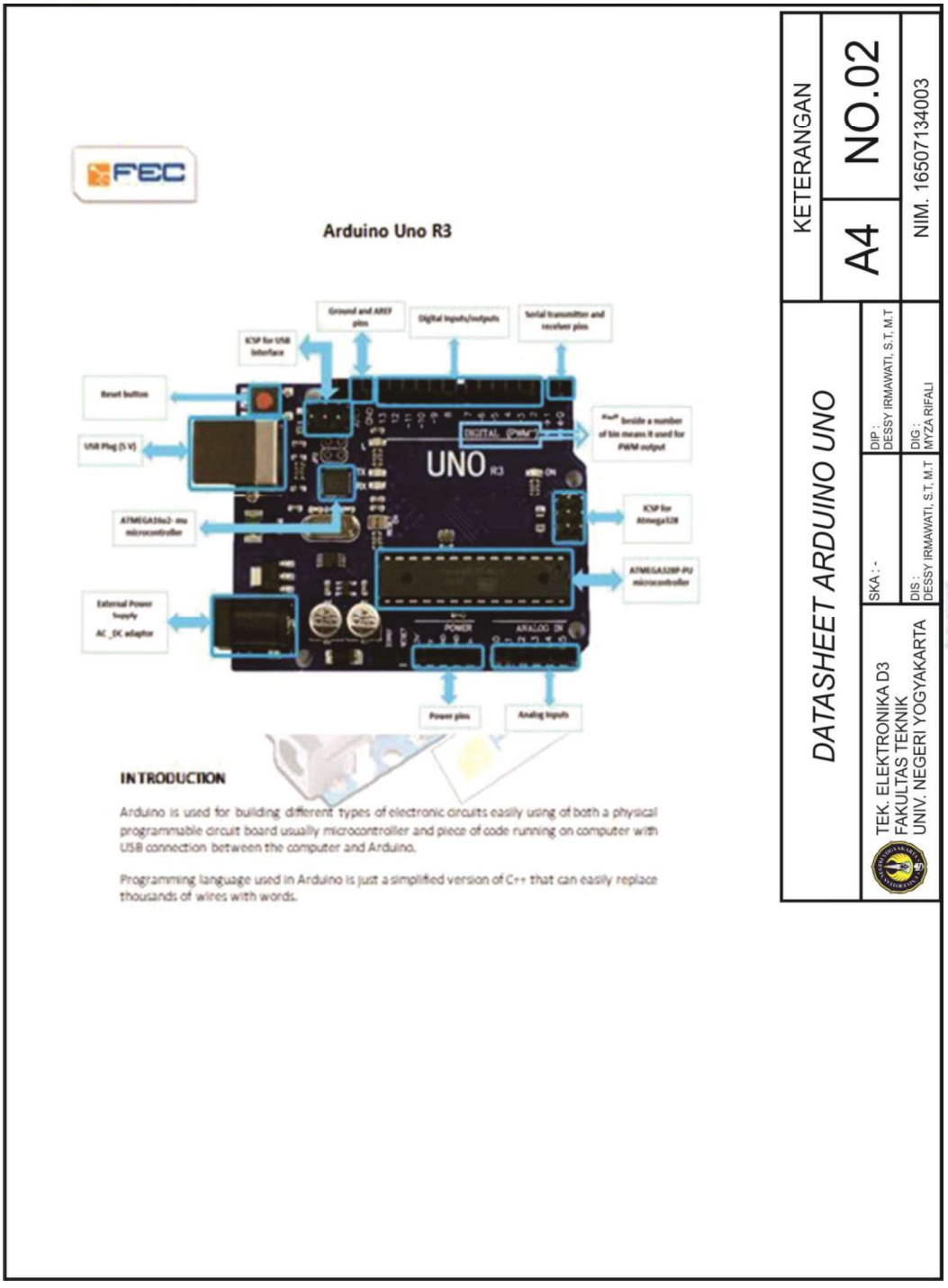
```
handles.output = hObject;
global state
state=0;
% Update handles structure
guidata(hObject, handles);

% UIWAIT makes coba6 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = coba6_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global val
delete(instrfindall);
x = serial('COM5');
set(x, 'BaudRate', 9600);
fopen(x);
%x=linspace(1,100);
%Baca data serial dalam format float

%val=zeros(512,1);
set(handles.edit4,'String','gagal merekam.')
tic;
for i=1:1280;
    val(i)= fscanf(x,'%d');
end
time=toc;

    %if ~isempty(instrfind);
    %out=instrfind(obj);
     fclose(x);
     save val
     plot(val)
     ylabel('Amplitudo');
     xlabel('Panjang sinyal');
     set(handles.edit4,'String','berhenti merekam')




% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global val
uisave val
```

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set (handles.edit2,'String','default');
set (handles.edit3,'String','default');
set (handles.edit4,'String','default');
cla(handles.axes1,'reset');

handles.val = [];
guidata(hObject,handles);
clear

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
homecoba6

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4 as a
double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a
double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
val=handles.val;
    CiriR=mean(handles.val);
    CiriVAR=var(handles.val);
    CiriSKEW=skewness(handles.val);
    CiriKURT=kurtosis(handles.val);
    CiriSD=std(handles.val);

x=[CiriR;CiriVAR;CiriSKEW;CiriKURT;CiriSD]

load net3
output=sim(net,x);
y=round(output);
        if y==1
          set(handles.edit2,'String','ABNORMAL')
        elseif y==2
          set(handles.edit2,'String','NORMAL')
        end



% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename,pathname]=uigetfile('*.mat*','pilih
file(mat)','MultiSelect','on');
if~isequal(filename,0)
in=load(filename);
set(handles.edit3,'String',filename
handles.in=in;
val=in.val;
handles.val=val;
guidata(hObject,handles)
axes(handles.axes1)
plot(handles.val)
xlabel('Panjang sinyal');
ylabel('Amplitudo');
else
    return
end
```

```
function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3 as a
double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global state
pilihan = questdlg('Ingin keluar dari aplikasi?',...
    'Keluar aplikasi','ya','tidak','ya');
switch pilihan
    case 'ya'
        state=0;
        delete(handles.figure1);
        %delete(handles.figure2);
        clear all;
    case 'tidak'
        return;
end
```

Lampiran 8. Datasheet Arduino



ARDUINO UNO R3 PHYSICAL COMPONENTS

ATMEGA328P-PU microcontroller

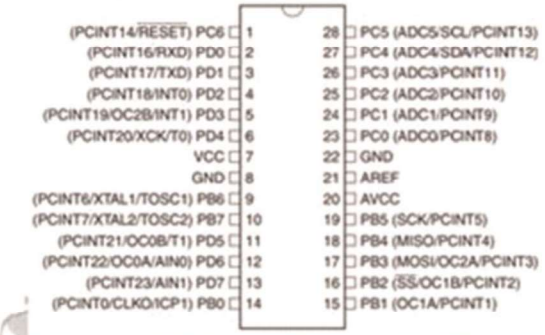
The most important element in Arduino Uno R3 is ATMEGA328P PU is an 8 bit Microcontroller with flash memory reach to 32k bytes. It's features as follow:

- **High Performance, Low Power AVR**
- **Advanced RISC Architecture**
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Up to 20 MIPS Throughput at 20 MHz
  - On chip 2 cycle Multiplier
- **High Endurance Non volatile Memory Segments**
  - 4/4/16/32K Bytes of In System Self Programmable Flash program memory
  - 256/512/512/1K Bytes EEPROM
  - 512/1K/1K/2K Bytes Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C
  - 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

- **Peripheral Features**
  - Two 8 bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16 bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8 channel 10 bit ADC in TQFP and QFN/MLF package
  - Temperature Measurement
  - 6 channel 10 bit ADC in PDIP Package
  - Temperature Measurement
  - Programmable Serial USART

KETERANGAN		
A4	NO.02	
DATASHEET ARDUINO UNO	DIP : DESSY IRMAWATI, S.T, M.T	NIM. 16507134003
	SKA : -	DIG : MYZA RIFALI
TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	DIS : DESSY IRMAWATI, S.T, M.T	

- **Pin configuration**



ATMEGA16u2 mu microcontroller

is a 8 bit microcontroller used as USB driver in Arduino uno R3 it's features as follow:

- **High Performance, low Power AVR**

• Advanced RISC Architecture

- 125 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

• Non volatile Program and Data Memories

- 8K/16K/32K Bytes of In System Self Programmable Flash
- 512/512/1024 EEPROM
- 512/512/1024 Internal SRAM
- Write/Erase Cycles: 10,000 Flash/ 100,000 EEPROM
Data retention: 20 years at 85°C/ 100 years at 25°C

KETERANGAN		
A4	NO.02	
DATASHEET ARDUINO UNO		NIM. 16507134003
TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA : - DIS : DESSY IRMAWATI, S.T. M.T	DIP : DESSY IRMAWATI, S.T. M.T DIG : MYZA RIFALI

- Optional Boot Code Section with Independent Lock Bits
- In-System Programming by on-chip Boot Program hardware activated after reset
- Programming Lock for Software Security

• USB 2.0 Full-speed Device Module with Interrupt on Transfer Completion

- Complies fully with Universal Serial Bus Specification REV 2.0
- 48 MHz PLL for Full-speed Bus Operation: data transfer rates at 12 Mbit/s
- Fully independent 176 bytes USB DPRAM for endpoint memory allocation
- Endpoint 0 for Control Transfers: from 8 up to 64-bytes
- 4 Programmable Endpoints:
  - IN or Out Directions
  - Bulk, Interrupt and Isochronous Transfers
  - Programmable maximum packet size from 8 to 64 bytes
  - Programmable single or double buffer
- Suspend/Resume Interrupts
- Microcontroller reset on USB Bus Reset without detach
- USB Bus Disconnection on Microcontroller Request

• Peripheral Features

- One 8-bit Timer/Counters with Separate Prescaler and Compare Mode (two 8-bit PWM channels)
- One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Mode(three 8-bit PWM channels)
- USART with SPI master only mode and hardware flow control (RTS/CTS)
- Master/Slave SPI Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

• On Chip Debug Interface (debugWIRE)

• Special Microcontroller Features

- Power-On Reset and Programmable Brown-out Detection
- Internal Calibrated Oscillator
- External and Internal Interrupt Sources
- Five Sleep Modes: Idle, Power-save, Power-down, Standby, and Extended Standby

• I/O and Packages

- 22 Programmable I/O Lines
- QFN32 (5x5mm) / TQFP32 packages

KETERANGAN

A4 NO.02

NIM. 16507134003

DATASHEET ARDUINO UNO

DIP : DESSY IRMAWATI, S.T, M.T

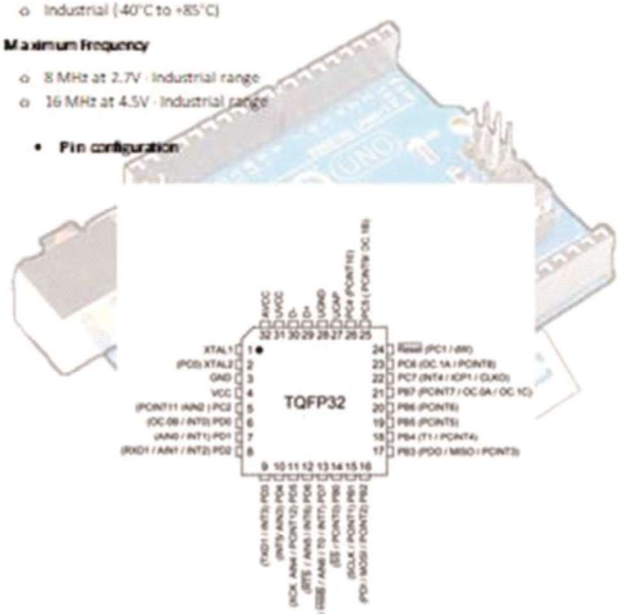
DIG : MYZA RIFALI

SKA : -

DIS : DESSY IRMAWATI, S.T, M.T

TEK. ELEKTRONIKA D3
FAKULTAS TEKNIK
UNIV. NEGERI YOGYAKARTA

- Operating Voltages
  - 2.7 - 5.5V
- Operating temperature
  - Industrial (-40°C to +85°C)
- Maximum Frequency
  - 8 MHz at 2.7V - Industrial range
  - 16 MHz at 4.5V - Industrial range
- Pin configuration



KETERANGAN

A4 | NO.02

NIM. 16507134003

DATASHEET ARDUINO UNO

DIP : DESSY IRMAWATI, S.T, M.T

DIG : MYZA RIFALI

SKA : -

DIS : DESSY IRMAWATI, S.T, M.T

TEK. ELEKTRONIKA D3
FAKULTAS TEKNIK
UNIV. NEGERI YOGYAKARTA

OTHER ARDUINO UNO R3 PARTS

Input and Output

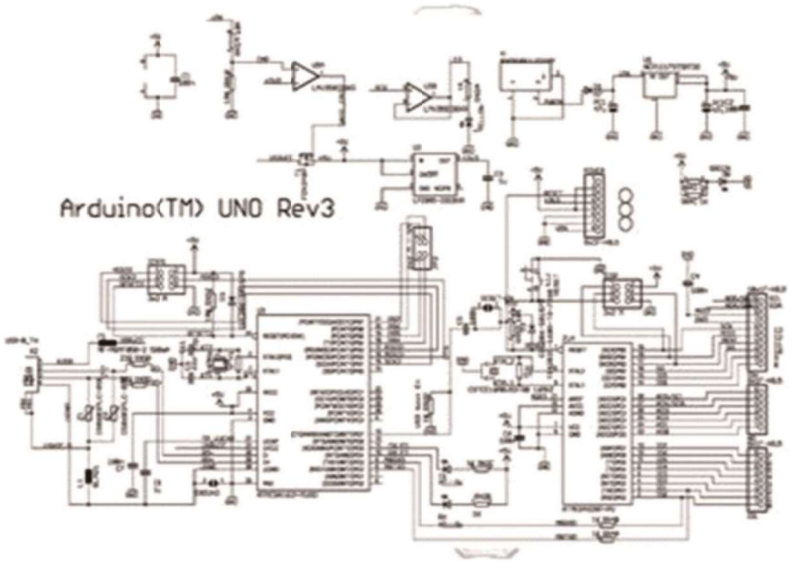
Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode (), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull up resistor (disconnected by default) of 20 50 k Ohms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB to TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8 bit PWM output with the analogWrite() function. o SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e.1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:

- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- AREF: Reference voltage for the analog inputs. Used with analogReference().
- Reset: Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

DATASHEET ARDUINO UNO	KETERANGAN		
	A4	NO.02	
TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA : -	DIP : DESSY IRMAWATI, S.T, M.T	NIM. 16507134003
	DIS : DESSY IRMAWATI, S.T, M.T	DIG : MYZA RIFALI	



ARDUINO UNO R3 SCHEMATIC DIAGRAM

DATASHEET ARDUINO UNO			KETERANGAN
			A4 / NO.02
TEK. ELEKTRONIKA D3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA : -	DIP : DESSY IRMAWATI, S.T, M.T	
	DIS : DESSY IRMAWATI, S.T, M.T	DIG : MYZA RIFALI	NIM. 16507134003

Lampiran 9. Datasheet modul AD8232

ANALOG DEVICES

Single-Lead, Heart Rate Monitor Front End

Data Sheet AD8232

FEATURES

Fully integrated single-lead ECG front end
Low supply current: 170 µA (typical)
Common-mode rejection ratio: 80 dB (dc to 60 Hz) Two or three electrode configurations
High signal gain (G = 100) with dc blocking capabilities 2-pole adjustable high-pass filter
Accepts up to ±300 mV of half cell potential
Fast restore feature improves filter settling
Uncommitted op amp
3-pole adjustable low-pass filter with adjustable gain
Leads off detection: ac or dc options
Integrated right leg drive (RLD) amplifier
Single-supply operation: 2.0 V to 3.5 V
Integrated reference buffer generates virtual ground
Rail-to-rail output
Internal RFI filter
8 kV HBM ESD rating
Shutdown pin
20-lead 4 mm × 4 mm LFCSP package

APPLICATIONS

Fitness and activity heart rate monitors
Portable ECG
Remote health monitors
Gaming peripherals
Biopotential signal acquisition

FUNCTIONAL BLOCK DIAGRAM

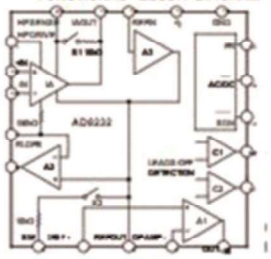


Figure 1.

GENERAL DESCRIPTION

The AD8232 is an integrated signal conditioning block for ECG and other biopotential measurement applications. It is designed to extract, amplify, and filter small biopotential signals in the presence of noisy conditions, such as those created by motion or remote electrode placement. This design allows for an ultralow power analog-to-digital converter (ADC) or an embedded microcontroller to acquire the output signal easily.

The AD8232 can implement a two-pole high-pass filter for eliminating motion artifacts and the electrode half-cell potential. This filter is tightly coupled with the instrumentation architecture of the amplifier to allow both large gain and high-pass filtering in a single stage, thereby saving space and cost.

An uncommitted operational amplifier enables the AD8232 to create a three-pole low-pass filter to remove additional noise. The user can select the frequency cutoff of all filters to suit different types of applications.

To improve common-mode rejection of the line frequency in the system and other undesired interferences, the AD8232 includes an amplifier for driven lead applications, such as right leg drive (RLD).

The AD8232 includes a fast restore function that reduces the duration of otherwise long settling tails of the high-pass filters. After an abrupt signal change that rails the amplifier (such as a leads off condition), the AD8232 automatically adjusts to a higher filter cutoff. This feature allows the AD8232 to recover quickly, and therefore, to take valid measurements soon after connecting the electrodes to the subject.

The AD8232 is available in a 4 mm × 4 mm, 20-lead LFCSP package. Performance is specified from 0°C to 70°C and is operational from −40°C to +85°C.

Rev. A Document Feedback

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 ©2012–2013 Analog Devices, Inc. All rights reserved.
Technical Support www.analog.com

KETERANGAN

A4 NO.03

NIM. 16507134003

DATASHEET MODUL AD8232

DIP : CESSY IRMAWATI, S.T, M.T

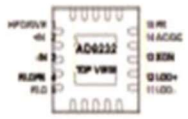
DIG : MYZA RIFALI

SKA : -

DIS : DESSY IRMAWATI, S.T, M.T

TEK. ELEKTRONIKA D3
FAKULTAS TEKNIK
UNIV. NEGERI YOGYAKARTA

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NOTES
1. CONNECT THE EXPOSED PAD TO GND OR LEAVE UNCONNECTED.

Figure 2. Pin Configuration

Table 3. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	HPDRIVE	High-Pass Driver Output. Connect HPDRIVE to the capacitor in the first high-pass filter. The AD8232 drives this pin to keep HPSENSE at the same level as the reference voltage.
2	+IN	Instrumentation Amplifier Positive Input. +IN is typically connected to the left arm (LA) electrode.
3	−IN	Instrumentation Amplifier Negative Input. −IN is typically connected to the right arm (RA) electrode.
4	RLDFB	Right Leg Drive Feedback Input. RLDFB is the feedback terminal for the right leg drive circuit.
5	RLD	Right Leg Drive Output. Connect the driven electrode (typically, right leg) to the RLD pin.
6	SW	Fast Restore Switch Terminal. Connect this terminal to the output of the second high-pass filter.
7	OPAMP+	Operational Amplifier Noninverting Input.
8	REFOUT	Reference Buffer Output. The instrumentation amplifier output is referenced to this potential. Use REFOUT as a virtual ground for any point in the circuit that needs a signal reference.
9	OPAMP−	Operational Amplifier Inverting Input.
10	OUT	Operational Amplifier Output. The fully conditioned heart rate signal is present at this output. OUT can be connected to the input of an ADC.
11	LOD−	Leads Off Comparator Output. In dc leads off detection mode, LOD− is high when the electrode to −IN is disconnected, and it is low when connected. In ac leads off detection mode, LOD− is always low.
12	LOD+	Leads Off Comparator Output. In dc leads off detection mode, LOD+ is high when the +IN electrode is disconnected, and it is low when connected. In ac leads off detection mode, LOD+ is high when either the −IN or +IN electrode is disconnected, and it is low when both electrodes are connected.
13	$\overline{SDN}$	Shutdown Control Input. Drive $\overline{SDN}$ low to enter the low power shutdown mode.
14	AC/$\overline{DC}$	Leads Off Mode Control Input. Drive the AC/$\overline{DC}$ pin low for dc leads off mode. Drive the AC/$\overline{DC}$ pin high for ac leads off mode.
15	FR	Fast Restore Control Input. Drive FR high to enable fast recovery mode, otherwise, drive it low.
16	GND	Power Supply Ground.
17	+Vs	Power Supply Terminal.
18	REFIN	Reference Buffer Input. Use REFIN, a high impedance input terminal, to set the level of the reference buffer.
19	IAOUT	Instrumentation Amplifier Output Terminal.
20	HPSENSE	High-Pass Sense Input for Instrumentation Amplifier. Connect HPSENSE to the junction of R and C that sets the corner frequency of the dc blocking circuit.
	EP	Exposed Pad. Connect the exposed pad to GND or leave it unconnected.

KETERANGAN

A4 | NO.03

NIM. 16507134003

DATASHEET MODUL AD8232

DIP : DESSY IRMAWATI, S.T, M.T

DIG : MYZA RIFALI

SKA : -

DIS : DESSY IRMAWATI, S.T, M.T

TEK. ELEKTRONIKA D3
FAKULTAS TEKNIK
UNIV. NEGERI YOGYAKARTA