Implementation

**Coding**

**Sample for login**

Private Sub cmdCancle\_Click()  
End  
End Sub  
Private Sub cmdOk\_Click()  
If txtPassword.Text = "Betty" Then  
Loginform.Hide  
MDIForm1.Show  
Else  
MsgBox "invalid password", vbOKOnly, "alert"  
txtname.Text = ""  
txtPassword.Text = ""  
txtname.SetFocus  
End If  
End Sub

**Coding of Bill Calculation Form:**

Dim u, u1, u2 As Long  
Dim X, Y, z, p, d, f As String  
Private Sub cmd1exit\_Click()  
Unload Me  
End Sub  
Private Sub cmdAdd\_Click()  
adodesc.Recordset.AddNew  
bmtxt.SetFocus  
End Sub  
Private Sub cmdCancle\_Click()  
bmtxt.Text = ""  
bdtxt.Text = ""  
tbtxt.Text = ""  
bntxt.Text = ""  
bntxt.Text = ""  
pdtxt.Text = ""  
txtbp.Text = ""  
End Sub  
Private Sub cmdclear\_Click()  
txtfix.Text = ""  
txted.Text = ""  
txttotal.Text = ""  
txted.Text = ""  
txtsc.Text = ""  
txtec.Text = ""  
txtunit.Text = ""  
tbtxt.Text = ""  
txtunit.SetFocus  
End Sub  
Private Sub cmdDelete\_Click()  
Dim res As String  
res = MsgBox("u w to delete the rec!", vbYesNo, "alert")  
If res = vbYes Then  
adodesc.Recordset.Delete  
MsgBox "record is deleted"  
adodesc.Refresh  
Else  
adodesc.Refresh  
End If  
End Sub  
Private Sub cmdExit\_Click()  
bmtxt.SetFocus  
End Sub  
Private Sub cmdFilter\_Click()  
Dim res  
res = InputBox("Enter The Criteria", "Filter")  
adodesc.RecordSource = "select \* from Billdec where" & res  
End Sub  
Private Sub cmdFind\_Click()  
Dim res As String  
res = InputBox("enter the criteria for a rec", "find")  
adodesc.Recordset.Find res, , adSearchForward, 1  
End Sub  
Private Sub cmdOk\_Click()  
If txtunit.Text = "" Then  
MsgBox "please enter unit"  
txtunit.Text = 0  
End If  
txtfix.Text = "Rs 60.00"  
f = 60  
u = CInt(txtunit.Text)  
If u <= 50 Then  
X = u \* 1.3  
txtec.Text = "Rs " & X  
txtsc.Text = "Rs " & CStr(u \* 0.1)  
p = u \* 0.1  
d = (X \* 14) / 100  
txted.Text = "Rs " & d  
txttotal.Text = "Rs " & (X + p + f + d)  
ElseIf (u > 50 And u <= 300) Then  
X = 50 \* 1.3  
u1 = u - 50  
Y = u1 \* 3  
txtec.Text = "Rs" & (X + Y)  
p = u \* 0.1  
txtsc.Text = "Rs" & p  
If u <= 100 Then  
d = ((X + Y) \* 15) / 100  
Else  
d = ((X + Y) \* 23) / 100  
End If  
txted.Text = "Rs" & d  
txttotal.Text = "Rs" & (X + Y + p + d + f)  
ElseIf u > 300 Then  
X = 50 \* 1.3  
Y = 250 \* 3  
z = (u - 300) \* 3.5  
txtec.Text = "Rs" & (X + Y + z)  
p = CStr(u \* 0.1)  
txtsc.Text = "Rs" & p  
If u <= 100 Then  
d = ((X + Y + z) \* 15) / 100  
Else  
d = ((X + Y + z) \* 23) / 100  
End If  
txted.Text = "Rs" & d  
txttotal.Text = "Rs" & (X + Y + z + p + d + f)  
End If  
End Sub  
Private Sub cmdoko\_Click()  
If txtbp.Text = "" Then  
tbtxt.Text = txttotal.Text  
Else  
tbtxt.Text = "Rs" & (X + Y + z + p + d + f) - Val(txtbp.Text)  
End If  
End Sub  
Private Sub cmdprevious\_Click()  
MDIForm1.Show  
Unload Me  
End Sub  
Private Sub cmdSave\_Click()  
Dim res As String  
res = MsgBox("u w to save the rec", vbYesNo + vbQuestion, "alert")  
If res = vbYes Then  
adodesc.Recordset.Update  
MsgBox "record is updated"  
'adodesc.Refresh  
Else  
adodesc.Recordset.CancelUpdate  
adodesc.Refresh  
End If  
End Sub  
Private Sub Form\_Activate()  
txtName.SetFocus  
End Sub  
Private Sub cmd1First\_Click()  
adocust.Recordset.MoveFirst  
cmd1First.Enabled = False  
cmd4Last.Enabled = True  
End Sub  
Private Sub cmd2Next\_Click()  
adocust.Recordset.MoveNext  
cmd1First.Enabled = True  
cmd4Last.Enabled = True  
If (adocust.Recordset.AbsolutePosition) = adPosEOF Then  
MsgBox ("no other record")  
adocust.Refresh  
End If  
End Sub  
Private Sub cmd3Previous\_Click()  
adocust.Recordset.MovePrevious  
cmd1First.Enabled = True  
cmd4Last.Enabled = True  
If adocust.Recordset.AbsolutePosition = adPosBOF Then  
MsgBox ("no other record")  
adocust.Refresh  
End If  
End Sub  
Private Sub cmd4Last\_Click()  
adocust.Recordset.MoveLast  
cmd1First.Enabled = True  
cmd4Last.Enabled = False  
End Sub

**Coding of Bill payment Form:**

Private Sub cmd1\_Click()  
adodesc.Recordset.MoveFirst  
cmd1.Enabled = False  
cmd4.Enabled = True  
adodesc.Refresh  
End Sub  
Private Sub cmd1First\_Click()  
adocust.Recordset.MoveFirst  
cmd1First.Enabled = False  
cmd4Last.Enabled = True  
End Sub  
Private Sub cmd2\_Click()  
Dim i As Integer  
If adocust.Recordset.Fields("cus\_id") = adodesc.Recordset.Fields("cus\_id") Then  
adodesc.Recordset.MoveNext  
cmd1.Enabled = True  
cmd4.Enabled = True  
End If  
If adodesc.Recordset.AbsolutePosition = adPosEOF Then  
MsgBox ("no other record")  
End If  
adodesc.Refresh  
End Sub  
Private Sub cmd2Next\_Click()  
adocust.Recordset.MoveNext  
cmd1First.Enabled = True  
cmd4Last.Enabled = True  
If adocust.Recordset.AbsolutePosition = adPosEOF Then  
MsgBox ("no other record")  
adocust.Refresh  
End If  
End Sub  
Private Sub cmd3\_Click()  
adodesc.Recordset.MovePrevious  
cmd1.Enabled = True  
cmd4.Enabled = True  
If adodesc.Recordset.AbsolutePosition = adPosBOF Then  
MsgBox ("no other record")  
adodesc.Refresh  
End If  
End Sub  
Private Sub cmd3Next\_Click()  
dombillcalfrm.Show  
Unload Me  
End Sub  
Private Sub cmd3Previous\_Click()  
adocust.Recordset.MovePrevious  
cmd1First.Enabled = True  
cmd4Last.Enabled = True  
If adocust.Recordset.AbsolutePosition = adPosBOF Then  
MsgBox "no other record"  
adocust.Refresh  
End If  
End Sub  
Private Sub cmd4\_Click()  
adodesc.Recordset.MoveLast  
cmd1.Enabled = True  
cmd4.Enabled = False  
End Sub  
Private Sub cmd4Last\_Click()  
adocust.Recordset.MoveLast  
cmd1First.Enabled = True  
cmd4Last.Enabled = False  
End Sub  
Private Sub cmdDelete\_Click()  
Dim res As String  
res = MsgBox("you want to delete the record", vbYesNo + vbQuestion, "alert")  
If res = vbYes Then  
adocust.Recordset.Delete  
adodesc.Recordset.Delete  
MsgBox "Record is Deleted"  
adocust.Refresh  
adodesc.Refresh  
Else  
adocust.Refresh  
adodesc.Refresh  
End If  
End Sub  
Private Sub cmdExit\_Click()  
Unload Me  
End Sub  
Private Sub cmdfirst\_Click()  
adodesc.Recordset.MoveFirst  
cmdfirst.Enabled = False  
cmdlast.Enabled = True  
End Sub  
Private Sub cmdnext\_Click()  
adodesc.Recordset.MoveNext  
cmdfirst.Enabled = True  
cmdlast.Enabled = True  
If adodesc.Recordset.AbsolutePosition = adPosEOF Then  
MsgBox ("no other record")  
adodesc.Refresh  
End If  
End Sub  
Private Sub cmdFilter\_Click()  
Dim res As String  
res = InputBox("Enter The Criteria", "Filter")  
'adodesc.Recordset.Fields("cus\_id") = adocust.Recordset.Fields("cus\_id")  
adocust.RecordSource = "select \* from bill where" & res  
'adodesc.RecordSource = "select billdec.billmonth,billdec.totalbill from billdec where" &  
res  
End Sub  
Private Sub cmdlastfive\_Click()  
DataGrid1.Visible = True  
Adodc1.RecordSource = "select top 5 totalbill,billmonth from billdec where cus\_id =" &  
Val(txtcusid)  
Adodc1.Refresh  
DataGrid1.Refresh  
End Sub  
Private Sub cmdNew\_Click()  
adocust.Recordset.AddNew  
adodesc.Recordset.AddNew  
txtname.SetFocus  
End Sub  
Private Sub cmdPrev\_Click()  
MDIForm1.Show  
Unload Me  
End Sub  
Private Sub cmdprevious\_Click()  
adodesc.Recordset.MovePrevious  
cmdfirst.Enabled = True  
cmdlast.Enabled = True  
If adodesc.Recordset.AbsolutePosition = adPosBOF Then  
MsgBox ("no other record")  
adodesc.Refresh  
End If  
End Sub  
Private Sub cmdlast\_Click()  
adodesc.Recordset.MoveLast  
cmdfirst.Enabled = True  
cmdlast.Enabled = False  
End Sub  
Private Sub cmdSave\_Click()  
Dim res As String  
res = MsgBox("you want to save the record", vbYesNo + vbQuestion, "alert")  
If res = vbYes Then  
adocust.Recordset.Update  
adodesc.Recordset.Fields("cus\_id") = adocust.Recordset.Fields("cus\_id")  
adodesc.Recordset.Update  
MsgBox "Record is Updated"  
adocust.Refresh  
adodesc.Refresh  
Else  
adocust.Recordset.CancelUpdate  
adodesc.Recordset.CancelUpdate  
adocust.Refresh  
adodesc.Refresh  
End If  
End Sub

**Hardware**

Firstly, the primary side of AMR system is explained. Current measurement transformer, which is used to measure the current with the power cable, is of the type ENTES ENT.B 75/5, 5VA, shown in Figure 2.



Figure 1. Current measurement transformer ENTES EN

Through the cable, it is connected to another current measurement transformer, which is exclusively designed for the Arduino board. Current transformer is based on TA12-100 current sensor that can measure current on the primary up to 5 A, and reduce it to 5 mA. The voltage is measured via voltage measuring transformers, which are directly linked to the Arduino board. Voltage measurement transformer is of a type ZMPT101B, shown in Figure.

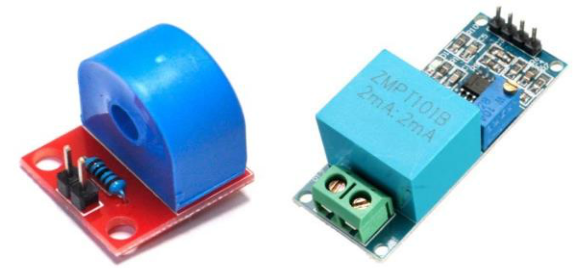


Figure 2. Current Measurement Transformer and Voltage measurement Transformer for the Arduino board

All primary equipment for measuring energy values of current (I) and voltage (U) have a relative error of 0.5 in relation to the measured value. Each measurement component as an output gives a sinusoidal shape of the analog signal, which is transmitted to the entrance of the Arduino board. All measuring sensors (I and U) are connected to the Arduino Uno R3 board that processes the signals. Signal processing involves the collection of signals from analog inputs on the board, calculation of the RMS value of the signal and the assignment of variables that are then suitable for sending the HTML 1.0. Figure shows an Arduino Uno R3 board.



Figure 3. Arduino Uno R3 board

\When the signals are encoded within the HTML code, they are forwarded then to a particular IP address with the help of ENC28J60 network module that this package of information through the TCP / IP protocol sends on ZYXEL 300 mbps router. Figure 5 shows the ENC28J60 network module for Arduino and ZYXEL router. Wi-Fi router is set to have an unlocked Wi-Fi network for the ease of managing experiment.



Figure 4. ENC28J60 network module for Arduino board and ZYXEL 300 mbps Wi-Fi router

The computer on which the server of applications is carried out for collecting information is HP laptop with more than 2 GB of RAM. Regardless of the platform that is used in the role of the server, application will be performed with no difficulties. In the event of a problem with hardware, one can simply switch application to a different hardware and resume operation.

.

Figure 5 HP laptop

Finally, all mentioned is connected to a cardboard plate and fixed inside a suitcase made of hard plastic. An important feature of all these components related to Arduino is a requirement of a power supply via the Arduino Uno board, which is powered from a source of direct current 9 V. All components are fastened with plastic laces, and all in the package is possible to transfer. Other components that were used are the board for joining electronic components, network cable with RJ45 interface, a thin 1.7 mm2 conductors for connecting components, adapters for supplying Arduino board and ZYXEL router from 9 V DC and 12 V DC, one extension cord for outdoor power. Figure 8 shows the entire HW system as closed and ready for transfer and open when it is in operation.

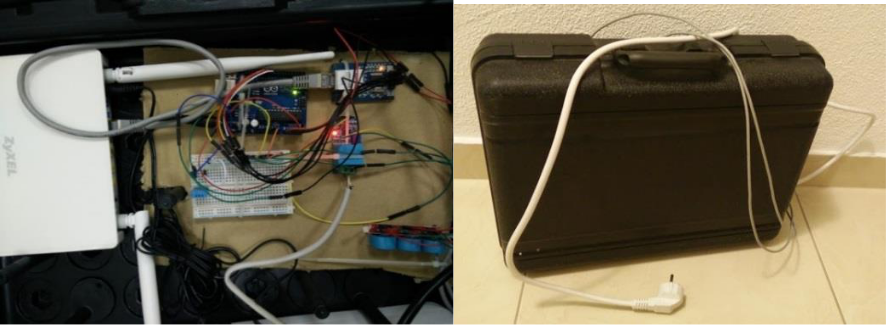
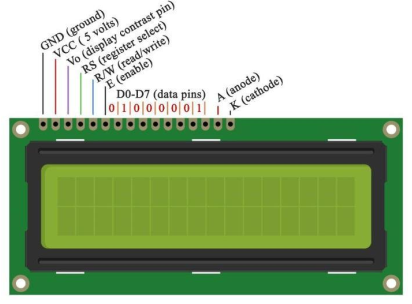


Figure 6. Left – opened bag with components in operation. Right – closed transfer suitcase

LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications. A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits. These modules are preferred over seven segments and other multi segment LEDs. The reasons being: LCDs are economical; easily programmable; have no limitation of displaying special even custom characters (unlike in seven segments), animations and so on.

Fig 6

**Software**

The first OSS that is used is the program for programming Arduino UNO R3 board version 1.6.9. In this software tool, all operations related to the work of the Arduino components are defined in the:

* method of collecting sensor signals,
* addressing signals to a variable,
* calibration of values,
* inclusion of libraries that are essential for the operation of the sensor
* ways of sending data to the output that the user selects (in this case the IP address). Figure 10 shows the programming interface for programming Arduino hardware.

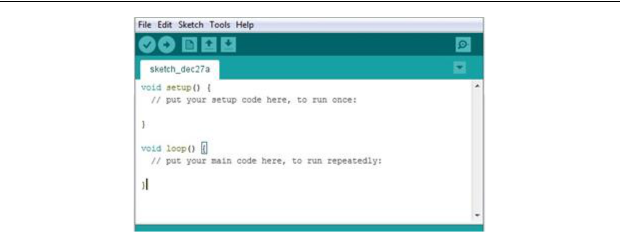


Figure 7. Application for programming Arduino UNO board

The following software that was used could have been any internet browser; in this case it was Google Chrome, which is available for free use. On the browser, the data is presented that are within the Wi-Fi network and are updated in real time. The Figure 11 shows Chrome with data presented from the meter.

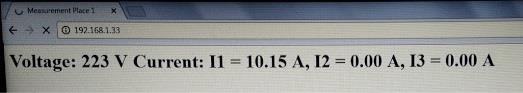


Figure 8. Google Chrome with data presentation from the meter

The operating system on which the server of application is carried out is Windows 10 version of 32bit. The programming environment in which the application is written for meter reading in Java is the Eclipse Neon for Windows. Figure 12 shows the programming environment of Eclipse Neon for application development in Java.

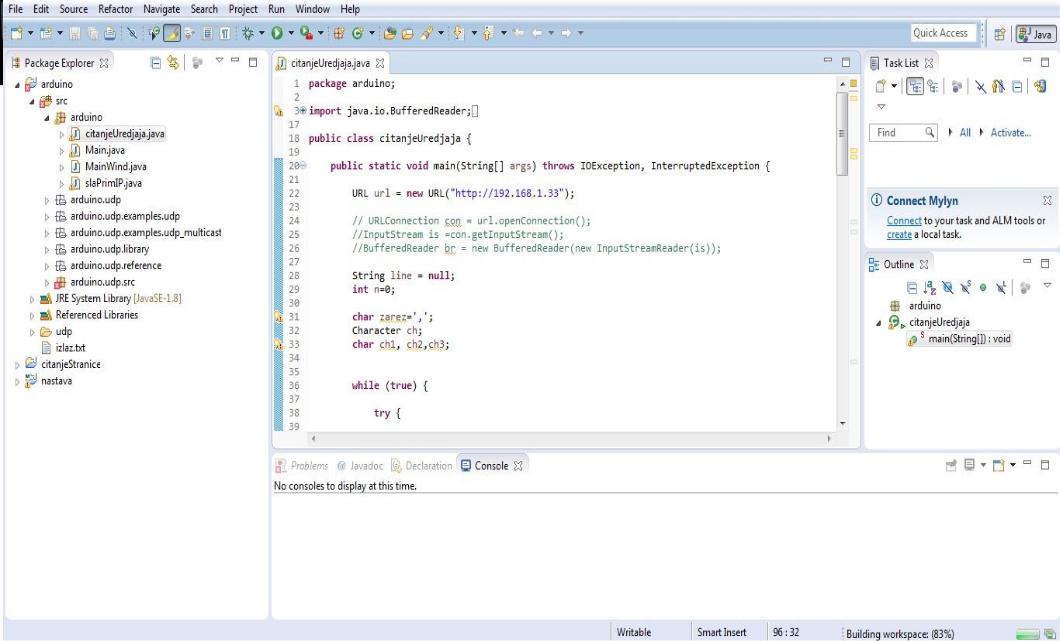


Figure 9. Eclipse Neon for development of JAVA application



The server of application reads the IP address with all values that are presented on it and stores them in a \* .txt file, that is later available for further data processing. Data from the IP address is read at time intervals of 200ms. The readings are far less common due to consumption of time required for calculating a signal from the sensor in RMS values that are transmitted to the IP address with the refresh time of 1s. Data transfer to a Wi-Fi bus bar takes about 1s. The speed processor used to run application has a delay of a few milliseconds. When all calculation is done, it yields a delay not less than 2 seconds. Usually, the delay may be from 2.5s to 20s. Figure shows the server of applications in the execution and .txt file in which data is stored.

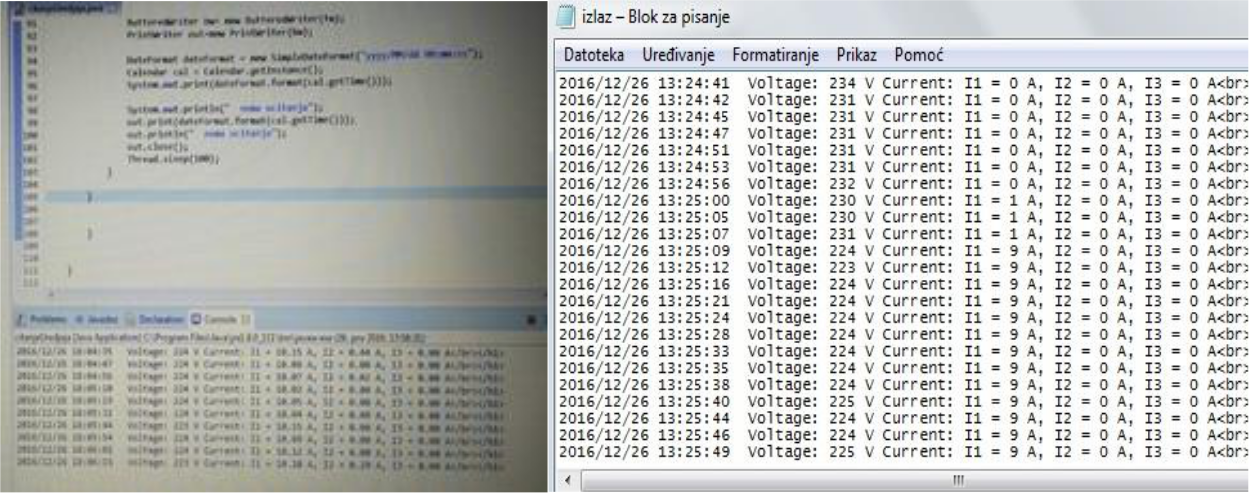


Figure 10. Left - server of applications when reading the ID address of the meter. Right - \*.txt file with data stored from the application

The flow of data through the architecture of AMR system with belonging formats is presented on Figure below.

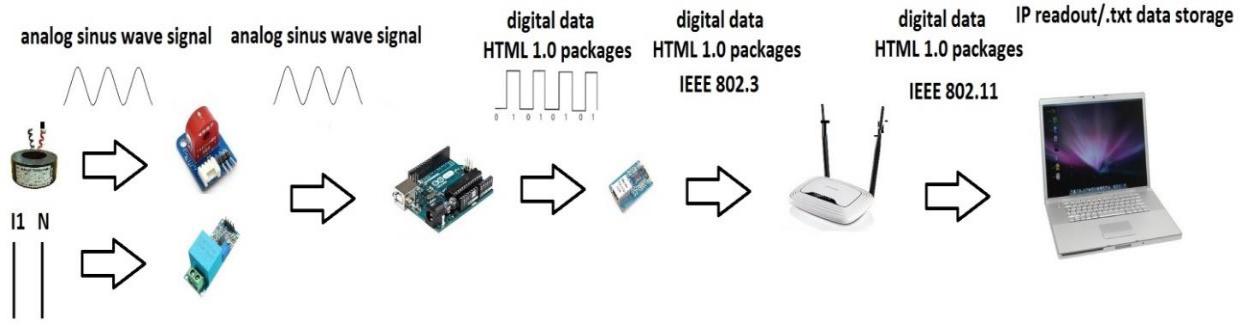


Figure 11. The flow of data through the architecture of AMR system

Extending the AMR system is done by attaching another set of measuring equipment with Arduino board to the ZYXEL router network cable. This involves one more IP address, which is again available for viewing in real time and recording with the application server.

Software has been used to model the energy meter. Not all components are not  
available in the software components library. The model of the system drawn using proteus is  
shown in Figure below

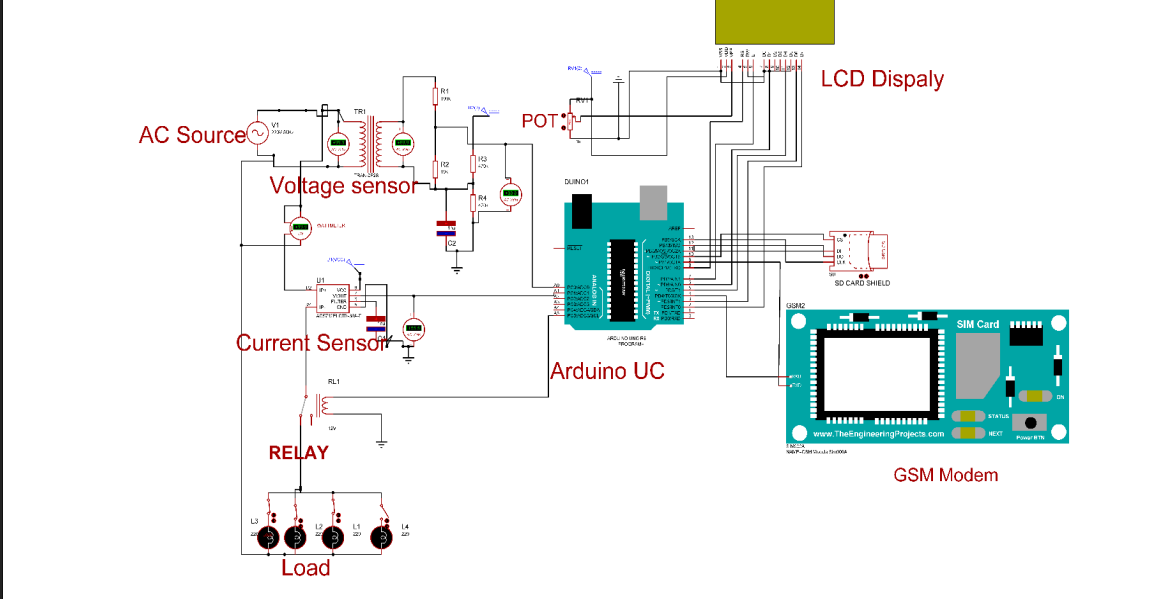
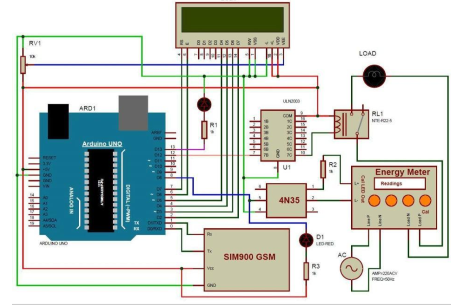


Fig 12

****

**Prototype Sample**

