LAMPIRAN I

CODING ARDUINO

```
//#include <OneWire.h>
#include <doxygen.h>
#include <NexButton.h>
#include <NexCheckbox.h>
#include <NexConfig.h>
#include <NexCrop.h>

#include <NexDualStateButton.h>
#include <NexGauge.h>

#include <NexGpio.h>
#include <NexHardware.h>
#include <NexHotspot.h>
#include <NexNumber.h>
#include <NexObject.h>
#include <NexPage.h>
#include <NexPicture.h>
#include <NexProgressBar.h>
#include <NexRadio.h>
#include <NexRtc.h>
#include <NexScrolltext.h>
#include <NexSlider.h>
#include <NexText.h>
#include <NexTimer.h>
#include <Nextion.h>
#include <NexTouch.h>
#include <NexUpload.h>
#include <NexVariable.h>
#include <NexWaveform.h>


#include <ModbusMaster.h>                // Load the (modified) library for modbus communication command codes. Kindly install at our website.

#define MAX485_DE      2                 // Define DE Pin to Arduino pin. Connect DE Pin of Max485 converter module to Pin 2 (default) Arduino board

#define MAX485_RE      3                 // Define RE Pin to Arduino pin. Connect RE Pin of Max485 converter module to Pin 3 (default) Arduino board
```

// These DE anr RE pins can be any other Digital Pins to be activated during transmission and reception process.

#define nexSer Serial2 // Controlling the Nextion HMI using Serial1 (pin18 of the Arduino Mega) to prevent interfering with code upload

#define dbgSer Serial // Debug using default Serial over USB towards Arduino Serial Monitor

#define DEBUG // Comment this out if you don't need to see what happens in the Serial Monitor

static uint8_t pzemSlaveAddr = 0x01; // Declare the address of device (meter) in term of 8 bits. You can change to 0x02 etc if you have more than 1 meter.

static uint16_t NewshuntAddr = 0x0001; // Declare your external shunt value. Default is 100A, replace to "0x0001" if using 50A shunt, 0x0002 is for 200A, 0x0003 is for 300A

// By default manufacturer may already set, however, to set manually kindly delete the "//" for line "// setShunt(0x01);" in void setup

ModbusMaster node; /* activate modbus master codes*/

float PZEMVoltage =0; /* Declare value for DC voltage */

float PZEMCurrent =0; /* Declare value for DC current*/

float PZEMPower =0; /* Declare value for DC Power */

float PZEMEnergy=0; /* Declare value for DC Energy */

unsigned long startMillisPZEM; /* start counting time for LCD Display */

unsigned long currentMillisPZEM; /* current counting time for LCD Display */

const unsigned long periodPZEM = 1000; // refresh every X seconds (in seconds) in LED Display. Default 1000 = 1 second

float vbatt;

int ResetEnergy = 0; /* reset energy function */

unsigned long startMillisEnergy; /* start counting time for LCD Display */

unsigned long currentMillisEnergy; /* current counting time for LCD Display */

const unsigned long periodEnergy = 1000; // refresh every X seconds (in seconds) in LED Display. Default 1000 = 1 second

float persen;

```
NexText t1 = NexText(1, 4, "t1"); //tegangan
NexText t3 = NexText(1, 6, "t3"); //arus
NexText t5 = NexText(1, 8, "t5"); //persen
NexNumber n0 = NexNumber(1, 12, "n0");
NexNumber n1 = NexNumber(1, 13, "n1");
NexProgressBar j0 = NexProgressBar(1,2,"j0");

NexWaveform graph = NexWaveform(1,10,"s0");


void setup()

{

  /*0 General*/


  Serial.begin(9600);                    /* to display readings in Serial
Monitor at 9600 baud rates */

  Serial1.begin(9600);
  nexSer.begin(9600);

  /* 1- PZEM-017 DC Energy Meter */


   setShunt(0x01);                       // Delete the "//" to set shunt rating
(0x01) is the meter address by default

//  resetEnergy(0x01);                   // By delete the double slash symbol,
the Energy value in the meter is reset. Can also be reset on the LCD
Display

  startMillisPZEM = millis();            /* Start counting time for run code

*/

  Serial3.begin(9600,SERIAL_8N2);        /* To assign communication
port to communicate with meter. with 2 stop bits (refer to manual)*/

                                         // By default communicate via Serial3 port:

pin 14 (Tx) and pin 15 (Rx)

  node.begin(pzemSlaveAddr, Serial3);    /* Define and start the
Modbus RTU communication. Communication to specific slave address
and which Serial port */

  pinMode(MAX485_RE, OUTPUT);            /* Define RE Pin as
Signal Output for RS485 converter. Output pin means Arduino command
the pin signal to go high or low so that signal is received by the
converter*/

  pinMode(MAX485_DE, OUTPUT);            /* Define DE Pin as
Signal Output for RS485 converter. Output pin means Arduino command
```

```
the pin signal to go high or low so that signal is received by the
converter*/

  digitalWrite(MAX485_RE, 0);                /* Arduino create output signal
for pin RE as LOW (no output)*/

  digitalWrite(MAX485_DE, 0);                /* Arduino create output signal
for pin DE as LOW (no output)*/

                                    // both pins no output means the converter is
in communication signal receiving mode

  node.preTransmission(preTransmission);      // Callbacks allow us to
configure the RS485 transceiver correctly

  node.postTransmission(postTransmission);

  //changeAddress(0XF8, 0x01);                // By delete the double slash
symbol, the meter address will be set as 0x01.

                                    // By default I allow this code to run every
program startup. Will not have effect if you only have 1 meter

  nexInit();
  delay(1000);

}


void loop()

{

   /* After Select button is pressed */
   if(ResetEnergy == 1)

   {

     currentMillisEnergy = millis();
     if(currentMillisEnergy - startMillisEnergy <= 5000)

/* if within 5 seconds <left> button is pressed, do reset energy*/

       {

         // Reset energy

           uint16_t u16CRC = 0xFFFF;

/* declare CRC check 16 bits*/

           static uint8_t resetCommand = 0x42;

/* reset command code*/

           uint8_t slaveAddr =0X01;
```

```
          u16CRC = crc16_update(u16CRC, slaveAddr);
          u16CRC = crc16_update(u16CRC, resetCommand);
          Serial.println("Resetting Energy");

          preTransmission();
/* trigger transmission mode*/
          Serial3.write(slaveAddr);
/* send device address in 8 bit*/
          Serial3.write(resetCommand);
/* send reset command */
          Serial3.write(lowByte(u16CRC));
/* send CRC check code low byte  (1st part) */
          Serial3.write(highByte(u16CRC));
/* send CRC check code high byte (2nd part) */
          delay(10);

          postTransmission();
/* trigger reception mode*/
          delay(100);

          while (Serial3.available())
/* while receiving signal from Serial3 from meter and converter */
            {
              Serial.print(char(Serial3.read()), HEX);
/* Prints the response and display on Serial Monitor (Serial)*/
              Serial.print(" ");

            }
        ResetEnergy=0;                                          /*
reset command switch back to default*/
/* go back to page 1 after reset*/
       }


     if(currentMillisEnergy - startMillisEnergy > 5000)                 /*
if more than 5 seconds <Left> button does not pressed, go back to main
page*/
       {
         ResetEnergy=0;                                         /*
```

```
reset command switch back to default*/

/* go back to page 1 after reset*/

      }

   }

  /* 1- PZEM-017 DC Energy Meter */
  currentMillisPZEM = millis();

/* count time for program run every second (by default)*/

  if (currentMillisPZEM - startMillisPZEM >= periodPZEM)

/* for every x seconds, run the codes below*/

  {

    uint8_t result;                                                    /*

Declare variable "result" as 8 bits */

    result = node.readInputRegisters(0x0000, 6);

/* read the 9 registers (information) of the PZEM-014 / 016 starting
0x0000 (voltage information) kindly refer to manual)*/

    if (result == node.ku8MBSuccess)

/* If there is a response */

      {

      uint32_t tempdouble = 0x00000000;

/* Declare variable "tempdouble" as 32 bits with initial value is 0 */

      PZEMVoltage = node.getResponseBuffer(0x0000) / 100.0;

/* get the 16bit value for the voltage value, divide it by 100 (as per
manual) */                                                          // 0x0000

to 0x0008 are the register address of the measurement value

      PZEMCurrent = node.getResponseBuffer(0x0001) / 100.0;

/* get the 16bit value for the current value, divide it by 100 (as per
manual) */


      if (PZEMCurrent <= 0.1){
        vbatt = 0;

        persen = 0;

        PZEMCurrent = 0;

      }

      else{
```

```
        vbatt =14.2-(PZEMCurrent*1000)*0.0004;
        persen = mapPecahan(vbatt, 11.2, 14.2, 0, 100);

      }

      tempdouble = (node.getResponseBuffer(0x0003) << 16) +
node.getResponseBuffer(0x0002);      /* get the power value. Power value
is consists of 2 parts (2 digits of 16 bits in front and 2 digits of 16 bits at
the back) and combine them to an unsigned 32bit */

      PZEMPower = tempdouble / 10.0;

/* Divide the value by 10 to get actual power value (as per manual) */


      tempdouble = (node.getResponseBuffer(0x0005) << 16) +
node.getResponseBuffer(0x0004);      /* get the energy value. Energy
value is consists of 2 parts (2 digits of 16 bits in front and 2 digits of 16
bits at the back) and combine them to an unsigned 32bit */

      PZEMEnergy = tempdouble;


//       Serial.print(PZEMVoltage, 1);

/* Print Voltage value on Serial Monitor with 1 decimal*/

//       Serial.print("V  ");

//       Serial.print(vbatt, 1);                                    /* Print
Voltage value on Serial Monitor with 1 decimal*/

//       Serial.print("Vbatt  ");

//       Serial.print(PZEMCurrent, 3);

//       Serial.print("A  ");

//       Serial.print(PZEMPower, 1);

//       Serial.print("W  ");

//       Serial.print(PZEMEnergy, 0);

//       Serial.print("Wh  ");

//       Serial.println();

      Serial.print(vbatt, 1);       Serial.print(",");

      Serial.print(PZEMCurrent, 2); Serial.print(",");
      Serial.print(persen);         Serial.println("");


//       persen = map(vbatt,11.2,14.3,0,100);
```

```
        Serial2.print(F("t1.txt=\""));
        Serial2.print(vbatt);
        Serial2.print(F("\""));
        Serial2.print(F("\xFF\xFF\xFF"));


        Serial2.print("n0.val=");
        Serial2.print(int(vbatt));
        Serial2.print(F("\xFF\xFF\xFF"));


        Serial2.print(F("t3.txt=\""));
        Serial2.print(PZEMCurrent);
        Serial2.print(F("\""));
        Serial2.print(F("\xFF\xFF\xFF"));


        Serial2.print("n1.val=");
        Serial2.print(int(PZEMCurrent));
        Serial2.print(F("\xFF\xFF\xFF"));


        Serial2.print("j0.val=");
        Serial2.print(int(persen));
        Serial2.print(F("\xFF\xFF\xFF"));


        Serial2.print(F("t5.txt=\""));
        Serial2.print(int(persen));
        Serial2.print(F("%\""));
        Serial2.print(F("\xFF\xFF\xFF"));


        if (pzemSlaveAddr==2)
/* just for checking purpose to see whether can read modbus*/
          {
            Serial.println();
          }
       }
        Else
          {
            Serial.println("Failed to read modbus");
          }
```

```
      startMillisPZEM = currentMillisPZEM ;

/* Set the starting point again for next counting time */

  }


}
```

CODING PZEM-017T

```
void preTransmission()
/* transmission program when triggered*/
{
  /* 1- PZEM-017 DC Energy Meter */
  digitalWrite(MAX485_RE, 1);
/* put RE Pin to high*/
  digitalWrite(MAX485_DE, 1);
/* put DE Pin to high*/
  delay(1);                                                    //
When both RE and DE Pin are high, converter is allow to transmit
communication
}


void postTransmission()
/* Reception program when triggered*/
{
  /* 1- PZEM-017 DC Energy Meter */
  delay(3);                                                    //
When both RE and DE Pin are low, converter is allow to receive
communication
  digitalWrite(MAX485_RE, 0);
/* put RE Pin to low*/
  digitalWrite(MAX485_DE, 0);
/* put DE Pin to low*/
}


void setShunt(uint8_t slaveAddr)
//Change the slave address of a node
{
```

```
  /* 1- PZEM-017 DC Energy Meter */
  static uint8_t SlaveParameter = 0x06;
/* Write command code to PZEM */
  static uint16_t registerAddress = 0x0003;
/* change shunt register address command code */


  uint16_t u16CRC = 0xFFFF;
/* declare CRC check 16 bits*/
  u16CRC = crc16_update(u16CRC, slaveAddr);
// Calculate the crc16 over the 6bytes to be send
  u16CRC = crc16_update(u16CRC, SlaveParameter);
  u16CRC = crc16_update(u16CRC, highByte(registerAddress));
  u16CRC = crc16_update(u16CRC, lowByte(registerAddress));
  u16CRC = crc16_update(u16CRC, highByte(NewshuntAddr));
  u16CRC = crc16_update(u16CRC, lowByte(NewshuntAddr));


  Serial.println("Change shunt address");
  preTransmission();
/* trigger transmission mode*/


  Serial3.write(slaveAddr);
/* these whole process code sequence refer to manual*/
  Serial3.write(SlaveParameter);
  Serial3.write(highByte(registerAddress));
  Serial3.write(lowByte(registerAddress));
  Serial3.write(highByte(NewshuntAddr));
  Serial3.write(lowByte(NewshuntAddr));
  Serial3.write(lowByte(u16CRC));
  Serial3.write(highByte(u16CRC));
  delay(10);
  postTransmission();
/* trigger reception mode*/
  delay(100);
  while (Serial3.available())
/* while receiving signal from Serial3 from meter and converter */


    {
      Serial.print(char(Serial3.read()), HEX);
/* Prints the response and display on Serial Monitor (Serial)*/
      Serial.print(" ");
    }
}
```

```
void changeAddress(uint8_t OldslaveAddr, uint8_t NewslaveAddr)
//Change the slave address of a node
{


  /* 1- PZEM-017 DC Energy Meter */


  static uint8_t SlaveParameter = 0x06;
/* Write command code to PZEM */
  static uint16_t registerAddress = 0x0002;
/* Modbus RTU device address command code */
  uint16_t u16CRC = 0xFFFF;
/* declare CRC check 16 bits*/
  u16CRC = crc16_update(u16CRC, OldslaveAddr);
// Calculate the crc16 over the 6bytes to be send
  u16CRC = crc16_update(u16CRC, SlaveParameter);
  u16CRC = crc16_update(u16CRC, highByte(registerAddress));
  u16CRC = crc16_update(u16CRC, lowByte(registerAddress));
  u16CRC = crc16_update(u16CRC, highByte(NewslaveAddr));
  u16CRC = crc16_update(u16CRC, lowByte(NewslaveAddr));


  Serial.println("Change Slave Address");
  preTransmission();
/* trigger transmission mode*/


  Serial3.write(OldslaveAddr);
/* these whole process code sequence refer to manual*/
  Serial3.write(SlaveParameter);
  Serial3.write(highByte(registerAddress));
  Serial3.write(lowByte(registerAddress));
  Serial3.write(highByte(NewslaveAddr));
  Serial3.write(lowByte(NewslaveAddr));


  Serial3.write(lowByte(u16CRC));
  Serial3.write(highByte(u16CRC));
  delay(10);
  postTransmission();
/* trigger reception mode*/
  delay(100);
```

```
  while (Serial3.available())
/* while receiving signal from Serial3 from meter and converter */
    {
      Serial.print(char(Serial3.read()), HEX);
/* Prints the response and display on Serial Monitor (Serial)*/
      Serial.print(" ");
    }
}


float mapPecahan(float x, float fromLow, float fromHigh, float
toLow, float toHigh)
{
  return (x - fromLow) * (toHigh - toLow) / (fromHigh - fromLow)
+ toLow;
}
```

LAMPIRAN II

KEMENTERIAN, PENDIDIKAN KEBUDAYAAN
RISET DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 fax. 0711-355918
Website : www.polisriwijaya.ac.id **E-mail :** info@polsri.ac.id

REKOMENDASI UJIAN LAPORAN AKHIR (LA)

Pembimbing Laporan Akhir memberikan rekomendasi kepada,

Nama	:	Yulfa Sholika
NIM	:	062030321029
Jurusan/Program Studi	:	Teknik Elektro / D3 – Teknik Elektronika
Judul Laporan Akhir	:	Perancangan Pengisian Cepat Dan Efisiensi Pada Baterai Lithium-Ion Menggunakan *Control Fuzzy Real-Time*

Mahasiswa tersebut telah memenuhi persyaratan dan dapat mengikuti Ujian Laporan Akhir (LA) pada Tahun Akademik 2022/2023.

Palembang, 7-8-23

Pembimbing I,

5/8 2023

Selamat Muslimin, S.T.,M.Kom.
NIP. 197907222008011007

Pembimbing II,

Ekawati Prihatini, S.T.,M.T.
NIP. 197903102002122005

KEMENTERIAN RISET,TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama

Nama	Yulfa Sholika
NIM	062030321029
Jurusan	Teknik Elektro
Program Studi	DIII Teknik Elektronika

Pihak Kedua

Nama	Selamat Muslimin,ST.,M.Kom.
NIP	197907222008011007
Jurusan	Teknik Elektro
Program Studi	DIII Teknik Elektronika

Pada hari ini ..Kamis..... tanggal ..02/02/23..... telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 3 (tiga) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari ..... pukul 07.30.., tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Palembang, 02/02/23

Pihak Pertama,

(Yulfa Sholika)
NIM 062030321029

Pihak Kedua,

(Selamat Muslimin,ST.,M.Kom.)
NIP 197907222008011007

Mengetahui,
Ketua Jurusan

(Ir. Iskandar Lutfi.,M.T.)
NIP 196501291991031002

KEMENTERIAN RISET,TEKNOLOGI DAN PENDIDIKAN TINGGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

KESEPAKATAN BIMBINGAN LAPORAN AKHIR (LA)

Kami yang bertanda tangan di bawah ini,

Pihak Pertama

Nama	Yulfa Sholika
NIM	062030321029
Jurusan	Teknik Elektro
Program Studi	DIII Teknik Elektronika

Pihak Kedua

Nama	Ekawati Prihatini,ST.,M.T.
NIP	197903102002122005
Jurusan	Teknik Elektro
Program Studi	DIII Teknik Elektronika

Pada hari ini ...Jum'at..... tanggal ..10 Februari 2023.. telah sepakat untuk melakukan konsultasi bimbingan Laporan Akhir.

Konsultasi bimbingan sekurang-kurangnya 1 (satu) kali dalam satu minggu. Pelaksanaan bimbingan pada setiap hari .....fentatif..... pukul ....., tempat di Politeknik Negeri Sriwijaya.

Demikianlah kesepakatan ini dibuat dengan penuh kesadaran guna kelancaran penyelesaian Laporan Akhir.

Palembang, 10 Februari 2023

Pihak Pertama,

(Yulfa Sholika)
NIM 062030321029

Pihak Kedua,

(Ekawati Prihatini,ST.,M.T.)
NIP 197903102002122005

Mengetahui,
Ketua Jurusan

(Ir. Iskandar Lutfi.,M.T.)
NIP 196501291991031002

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN AKHIR

Nama : Yulfa Sholika
NIM : 062030321029
Jurusan/Program Studi : Teknik Elektro / DIII Teknik Elektronika
Judul Laporan Akhir : Perancangan Pengisian Cepat Dan Efisiensi Pada Baterai Lithium-Ion Menggunakan *Control Fuzzy Real-Time*

Pembimbing I : Selamat Muslimin,ST.,M.Kom

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	06/02/2023	-Pembahasan Konsultasi LA -Konsultasi referensi judul proposal	
2.	24/02/2023	Penentuan judul proposal	
3.	15/03/2023	Konsultasi Rumusan masalah yang akan di bahas.	
4.	17/03/2023	Bimbingan BAB 1, 2 proposal	
5.	28/03/2023	- Pembahasan dan Bimbingan BAB 3. - Acc BAB 1, 2 proposal	
6.	6/04/2023	- Revisi BAB 3 proposal	
7.	17/04/2023	Acc BAB 3 proposal	

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
8.	24 / 05 / 2023	Acc BAB 1, 2, 3, proposal	
9.	21 / Juni / 2023	Konsultasi Alat Charging	
10.	3 / 08 / 2023	- Pengujian Alat Charging - pengambilan data V, I, T pada charging	
11.	4 / 08 / 2023	Bimbingan Laporan Akhir	
12.	5 / 08 / 2023	- Acc Laporan Akhir - Rekomendasi sidang LA.	

Palembang, 5 September 2023

Ketua Jurusan/KPS,

(Ir. Iskandar Lutfi.,M.T.)
NIP 196501291991031002

Catatan:
*) melingkari angka yang sesuai.
Ketua Jurusan/Ketua Program Studi harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.
Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Akhir.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
8.	24 / 05 / 2023	Acc BAB 1, 2, 3, proposal	
9.	21 / Juni / 2023	Konsultasi Alat Charging	
10.	3 / 08 / 2023	- Pengujian Alat Charging - pengambilan data V, I, T pada charging	
11.	4 / 08 / 2023	Bimbingan Laporan Akhir	
12.	5 / 08 / 2023	- Acc Laporan Akhir - Rekomendasi Sidang LA.	

Palembang, 5 September 2023.

Ketua Jurusan/KPS,

(Ir. Iskandar Lutfi.,M.T.)
NIP 196501291991031002

Catatan:
*) melingkari angka yang sesuai.
Ketua Jurusan/Ketua Program Studi harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.
Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Akhir.

KEMENTERIAN PENDIDIKAN DAN KEBUDAYAAN
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

LEMBAR BIMBINGAN LAPORAN AKHIR

Nama : Yulfa Sholika
NIM : 062030321029
Jurusan/Program Studi : Teknik Elektro / DIII Teknik Elektronika
Judul Laporan Akhir : Perancangan Pengisian Cepat Dan Efisiensi Pada Baterai Lithium-Ion Menggunakan *Control Fuzzy Real-Time*

Pembimbing II : Ekawati Prihatini,ST.,M.T.

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
1.	24 Mei 2023	- Pengajuan judul proposal & rancangan Bab I - IV Prop.	
2.	31 Mei 2023	- Acc judul proposal. LA.	
3.	21 Juli 2023	- Predictive Table u/ data pada Bab IV	
4.	27 Juli 2023	- Revisi Analisa % eff daya u/ Fullcharge	
5.	31 Juli 2023	- Pengujian Alat di Lab, data V, I, T Charging	
6.	3 Agt 2023	- Acc Bab IV Analisa, Lengkapi daftar2	
7.	7 Agt 2023	- Rekomendasi Sidang LA.	

No.	Tanggal	Uraian Bimbingan	Tanda Tangan Pembimbing
8.			
9.			
10.			
11.			
12.			

Palembang, 28 Agustus 2023 .....

Ketua Jurusan/KPS,

(Ir. Iskandar Lutfi.,M.T.)
NIP 196501291991031002

Catatan:
*) melingkari angka yang sesuai.
Ketua Jurusan/Ketua Program Studi harus memeriksa jumlah pelaksanaan bimbingan sesuai yang dipersyaratkan dalam Pedoman Laporan Akhir sebelum menandatangani lembar bimbingan ini.
Lembar pembimbingan LA ini harus dilampirkan dalam Laporan Akhir.

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI SRIWIJAYA
Jalan Srijaya Negara, Palembang 30139
Telp. 0711-353414 Fax. 0711-355918
Website : www.polisriwijaya.ac.id E-mail : info@polsri.ac.id

PELAKSANAAN REVISI LAPORAN AKHIR

Mahasiswa berikut,

Nama	:	Yulfa Sholika
NIM	:	062030321029
Jurusan/Program Studi	:	Teknik Elektro/DIII Teknik Elektronika
Judul Laporan Akhir	:	Perancangan Pengisian Cepat Dan Efisiensi Pada Baterai Lithium-Ion Menggunkan *Control Fuzzy Real-Time*

Telah melaksanakan revisi terhadap Laporan Akhir yang diujikan pada hari Rabu tanggal 16 bulan Agustus tahun 2023 Pelaksanaan revisi terhadap Laporan Akhir tersebut telah disetujui oleh Dosen Penguji yang memberikan revisi:

No.	Komentar	Nama Dosen Penguji *)	Tanggal	Tanda Tangan
01	ACC	IR. A. RAHMAN, M.T	18/8-23	
02	Acc	EVELINA, S.T., M.Kom	24/8 23	
03	Acc	EKAWATI PRIHATINI, S.T., M.T	28/8-23	
04	ACC	JOHANSYAH ALRASYID, S.T,. M.Kom	28/8/2023	

Palembang, 18 Agustus 2023

Ketua Penguji **),

(IR. A. RAHMAN, M.T)
NIP : 196202051993031002

Catatan:
*) Dosen penguji yang memberikan revisi saat ujian laporan akhir.
**) Dosen penguji yang ditugaskan sebagai Ketua Penguji saat ujian LA.
Lembaran pelaksanaan revisi ini harus dilampirkan dalam Laporan Akhir.

LAMPIRAN III

PZEM-003/017 DC communication module

Overview

This document describes the specification of the PZEM-003/017 DC communication module, the module is mainly used for measuring DC voltage, current, active power, frequency and energy consumption, the module is without display function, the data is read through the RS485 interface.

PZEM-003: Measuring Range 10A (Built-in Shunt)

PZEM-017: Measuring Range 50A、100A、200A、300A (the current range is depend on the external shunt specification)

1. Function description

1.1 Voltage

1.1.1 Measuring range:0.05-300V. (when the test voltage is ＜7V, please use the independent power supply mode)

1.1.2 Resolution:0.01V.

1.1.3 Measurement accuracy:1%.

1.2 Current

1.2.1 Measuring range:0.01-10A(PZEM-003);0.02-300A(PZEM-017;can be matched with50、100、200、300A four kinds of shunt).

1.2.2 Resolution:0.01A

1.2.3 Measurement accuracy:1%

1.3 Power

1.3.1 Measuring range:0.1-3kW(PZEM-003);0.2-90kW(PZEM-017)

1.3.2 Resolution: 0.1W

1.3.3 Measurement accuracy:1%

1.4 Energy Consumption

1.4.1 Measuring range: 0-9999kWh

1.4.2 Resolution: 1Wh

1.4.3 Measurement accuracy:1%

1.4.4 Reset energy: use software to reset.

1.5 Over Voltage alarm

Voltage threshold can be set, divide into high voltage and low voltage threshold, when the measured voltage exceeds the threshold, it can alarm

The default high voltage threshold is 300V, the default low voltage threshold is 7V.

1.6 Communication interface

RS485 interface.

2. Communication protocol

2.1 Physical layer protocol

Physical layer use UART to RS485 communication interface.

Baud rate is 9600, 8 data bits, 2 stop bit, no parity.

2.2 Application layer protocol

The application layer use the Modbus-RTU protocol to communicate. At present, it only supports function codes such as 0x03 (Read Holding Register), 0x04 (Read Input Register), 0x06 (Write Single Register), 0x41 (Calibration), 0x42 (Reset energy).etc.

0x41 function code is only for internal use (address can be only 0xF8), used for factory calibration and return to factory maintenance occasions, after the function code to increase 16-bit password, the default password is 0x3721.

The address range of the slave is 0x01 ~ 0xF7. The address 0x00 is used as the broadcast address, the slave does not need to reply the master. The address 0xF8 is used as the general address, this address can be only used in single-slave environment and can be used for calibration etc.operation.

2.3 Read the measurement result

The command format of the master reads the measurement result is(total of 8 bytes):

Slave Address + 0x04 + Register Address High Byte + Register Address Low Byte + Number of Registers High Byte + Number of Registers Low Byte + CRC Check High Byte + CRC Check Low Byte.

The command format of the reply from the slave is divided into two kinds:

Correct Reply: Slave Address + 0x04 + Number of Bytes + Register 1 Data High Byte + Register 1 Data Low Byte + ... + CRC Check High Byte + CRC Check Low Byte

Error Reply: Slave address + 0x84 + Abnormal code + CRC check high byte + CRC check low byte

Abnormal code analyzed as following (the same below)

- 0x01,Illegal function;

- 0x02,Illegal address;
- 0x03,Illegal data;
- 0x04,Slave error.

The register of the measurement results is arranged as the following table

Register address	Description	Resolution
0x0000	Voltage value	1LSB correspond to 0.01V
0x0001	Current value	1LSB correspond to 0.01A
0x0002	Power value low 16 bits	1LSB correspond to 0.1W
0x0003	Power value high 16 bits	
0x0004	Energy value low 16 bits	1LSB correspond to 1Wh
0x0005	Energy value high 16 bits	
0x0006	High voltage alarm status	0xFFFF is alarm,0x0000 is not alarm
0x0007	Low voltage alarm status	0xFFFF is alarm,0x0000 is not alarm

For example, the master sends the following command (CRC check code is replaced by 0xHH and 0xLL, the same below):

0x01 + 0x04 + 0x00 + 0x00 + 0x00 + 0x08 + 0xHH + 0xLL

Indicates that the master needs to read 8 registers with slave address 0x01 and the start address of the register is 0x0000.

The correct reply from the slave is as following:

0x01 + 0x04 + 0x10 + 0x27 + 0x10 + 0x00 + 0x64 + 0x03 + 0xE8 + 0x00 + 0x00 + 0x00 + 0x00 + 0x00 + 0x00 + 0x00 + 0x00 + 0x00 + 0x00 + 0x00 + 0xHH + 0xLL

The above data shows

- Voltage is 0x2710，converted to decimal is 10000,display 100.00V；
- Current is 0x0064，converted to decimal is 100,display 1.00A；
- Power is 0x000003E8，converted to decimal is 1000,display 100.0W；
- Energy is 0x00000000，converted to decimal is 0,display 0Wh；
- High voltage alarm status 0x0000,indicates the current voltage is lower than the high voltage threshold.
- Low voltage alarm status 0x0000,indicates the current voltage is higher than the low voltage threshold.

2.4 Read and modify the slave parameters

At present,it only supports reading and modifying slave address and power alarm threshold

The register is arranged as the following table

Register address	Description	Resolution

0x0001	Low voltage alarm threshold （1~350V）,default is 7V	1LSB correspond to 0.01V
0x0002	Modbus-RTU address	The range is 0x0001~0x00F7
0x0003	The current range(only for PZEM-017)	0x0000：100A 0x0001：50A 0x0002：200A 0x0003：300A

The command format of the master to read the slave parameters and read the measurement results are same(described in details in Section 2.3), only need to change the function code from 0x04 to 0x03.

The command format of the master to modify the slave parameters is (total of 8 bytes):

Slave Address + 0x06 + Register Address High Byte + Register Address Low Byte + Register Value High Byte + Register Value Low Byte + CRC Check High Byte + CRC Check Low Byte.

The command format of the reply from the slave is divided into two kinds:

Correct Response: Slave Address + 0x06 + Number of Bytes + Register Address Low Byte + Register Value High Byte + Register Value Low Byte + CRC Check High Byte + CRC Check Low Byte.

Error Reply: Slave address + 0x86 + Abnormal code + CRC check high byte + CRC check low byte.

For example, the master sets the slave's high voltage alarm threshold:

0x01 + 0x06 + 0x00 + 0x00 + 0x4E + 0x20 + 0xHH + 0xLL

Indicates that the master needs to set the 0x0000 register (high voltage alarm threshold) to 0x4E20（200.00V）.

Set up correctly, the slave return to the data which is sent from the master.

For example, the master sets the low voltage alarm threshold of the slave

0x01 + 0x06 + 0x00 + 0x01 + 0x03 + 0xE8 + 0xHH + 0xLL

Indicates that the master needs to set the 0x0001 register (low voltage alarm threshold) to 0x03E8(10.00V).

Set up correctly, the slave return to the data which is sent from the master.

For example, the master sets the address of the slave

0x01 + 0x06 + 0x00 + 0x02 + 0x00 + 0x05 + 0xHH + 0xLL

Indicates that the master needs to set the 0x0002 register (Modbus-RTU address) to 0x0005

Set up correctly, the slave return to the data which is sent from the master.

2.5 Reset energy

The command format of the master to reset the slave's energy is (total 4 bytes):

Slave address + 0x42 + CRC check high byte + CRC check low byte.

Correct reply: slave address + 0x42 + CRC check high byte + CRC check low byte.

Error Reply: Slave address + 0xC2 + Abnormal code + CRC check high byte + CRC check low byte

2.6 Calibration

The command format of the master to calibrate the slave is (total 6 bytes):

0xF8 + 0x41 + 0x37 + 0x21 + CRC check high byte + CRC check low byte.

Correct reply: 0xF8 + 0x41 + 0x37 + 0x21 + CRC check high byte + CRC check low byte.

Error Reply: 0xF8 + 0xC1 + Abnormal code + CRC check high byte + CRC check low byte.

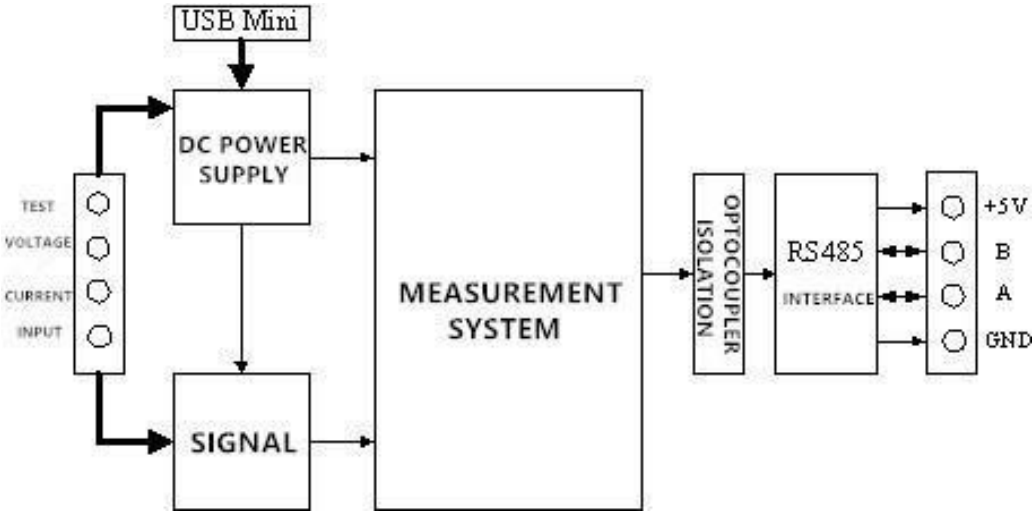
It should be noted that the calibration takes 3 to 4 seconds, after the master sends the command, if the calibration is successful, it will take 3 ~ 4 seconds to receive the response from the slave.

2.7 CRC check

CRC check use 16bits format, occupy two bytes, the generator polynomial is X16 + X15 + X2 +1, the polynomial value used for calculation is 0xA001.

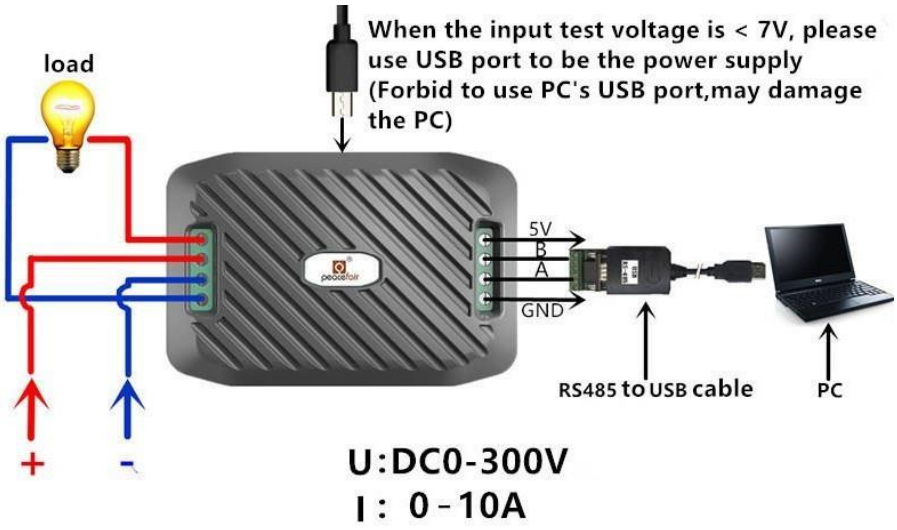
The value of the CRC check is all results of a frame data checking divide CRC

3. Functional block diagram

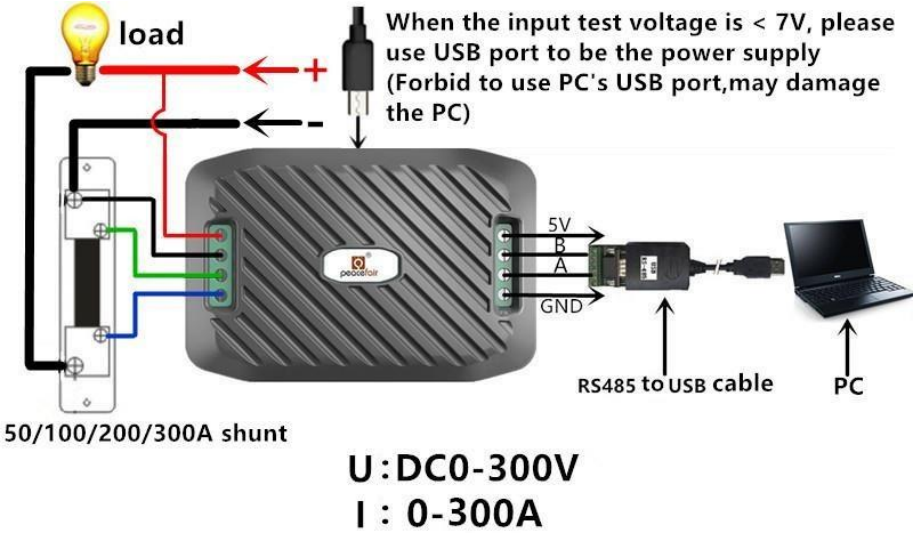


Picture 3 Functional block diagram

4. Wiring diagram



Picture 4.1 PZEM-003Wiring diagram



Picture 4.2 PZEM-017 Wiring diagram

5. Other instructions

5.1 RS485 interface is passive output, need external connect 5V power supply and the the external power supply should ＞100mA.

5.2 When the input test voltage is less than 7V, it must supply 5V independent work voltage through MICRO USB port；

5.3 Working temperature

-20’C ~ +60’C。

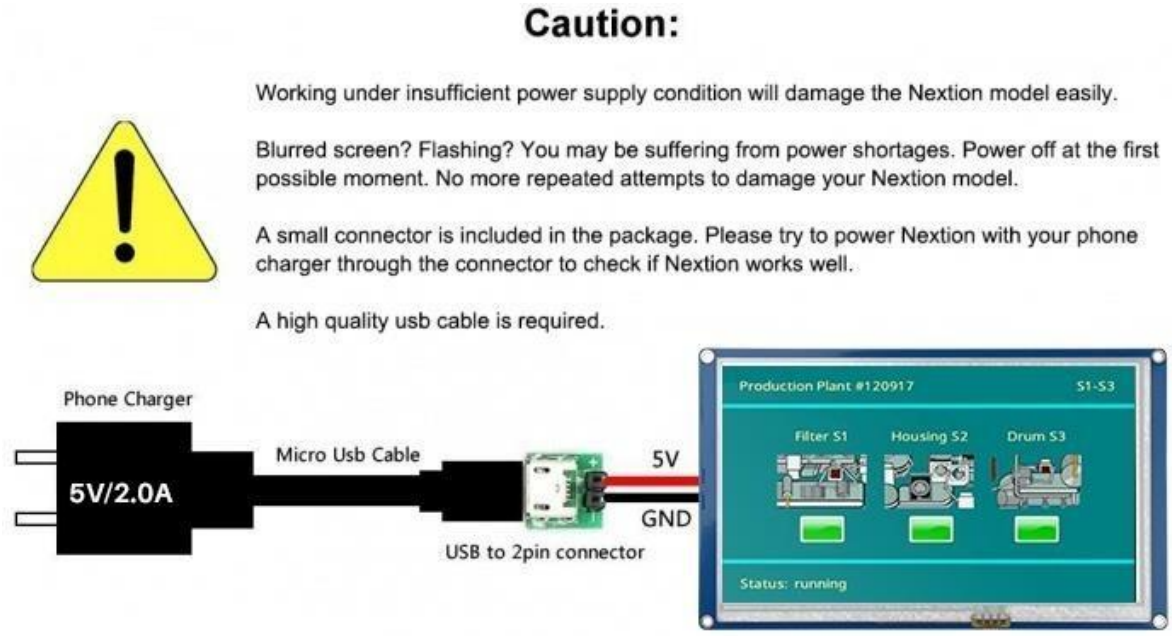
NX8048T070

Overview

Nextion is a seamless Human Machine Interface (HMI) solution that provides a control and visualisation interface between a human and a process, machine, application or appliance. Nextion is mainly applied to IoT or consumer electronics field. It is the best solution to replace the traditional LCD and LED Nixie tube. With the Nextion Editor software (Official Download), users are able to create and design their own interfaces forNextion display.

Package includes: Nextion Display, connecting wire, a power supply test board. Go Shopping: NX8048T070 (IM150416007)

Note: the small power supply test board and connecting wire inside the package allowyou to test if the electrical supply is enough or not. See the image below on how to useit.



Nextion Models

Nextion Type	Basic Series
Nextion Models	NX8048T070_011N (N: No touch)
	NX8048T070_011R (R: Resistive touchscreen)