LAMPIRAN

Lampiran 1 : Datasheet Arduino Uno

Overview

The Arduino Uno is a microcontroller board based on the ATmega328 (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATMega16U2 (Atmega8L up to Teensy 3.2) programmed as a USB-to-serial converter.

Revision 2 of the Uno board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Revision 3 of the board has the following new features:

- 1.0 pinout: added SDA and SCL pins that are near to the AREF pin and two other new pins placed near to the RESET pin, the IOREF that allow the shields to adapt to the voltage provided from the board. In future, shields will be compatible both with the board that use the AVR, which operate with 5V and with the Arduino Due that operate with 3.3V. The second one is a not connected pin, that is reserved for future purposes.
- Stronger RESET circuit.
- Atmega 16U2 replace the 8U2.

"Uno" means one in Italian and is named to mark the upcoming release of Arduino 1.0. The Uno and version 1.0 will be the reference versions of Arduino, moving forward. The Uno is the latest in a series of USB Arduino boards, and the reference model for the Arduino platform; for a comparison with previous versions, see the index of Arduino boards.

Summary

<table>
<thead>
<tr>
<th>Microcontroller</th>
<th>ATmega328</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>5V</td>
</tr>
<tr>
<td>Input Voltage (recommended)</td>
<td>7-12V</td>
</tr>
</tbody>
</table>
Input Voltage (limits) 6-20V
Digital I/O Pins 14 (of which 6 provide PWM output)
Analog Input Pins 6
DC Current per I/O Pin 40 mA
DC Current for 3.3V Pin 50 mA
Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM 2 KB (ATmega328)
EEPROM 1 KB (ATmega328)
Clock Speed 16 MHz

Schematic & Reference Design

EAGLE files: arduino-uno-Rev3-reference-design.zip (NOTE: works with Eagle 6.0 and newer)
Schematic: arduino-uno-Rev3-schematic.pdf

Note: The Arduino reference design can use an Atmega8, 168, or 328. Current models use an ATmega328, but an Atmega8 is shown in the schematic for reference. The pin configuration is identical on all three processors.

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-watt) or battery. The adapter can be connected by plugging a 2.1mm outer, 5.5mm inner plug into the board’s power jack. Leads from a battery can be inserted in the GND and VIN pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than 5 volts and the board may not work. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts. The power pins are as follows:

- VIN. The input voltage to the Arduino board when it’s using an external power source (as opposed to 5 volts from the USB connection, or another regulated power source). You can supply voltage through this pin, or input voltage via the power jack, access it through this pin.
- 5V. This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7-12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don’t recommend it.
- 3.3V. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

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 kOhms. 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. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
• **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 it is 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**.

There are a couple of other pins on the board:

• **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.

See also the mapping between Arduino pins and ATmega328 ports. The mapping for the ATmega8, 168, and 328 is identical.

**Communication**

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TX (RX) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual COM port to software on the computer. The '16U2 firmware uses the standard USB COM ports, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino. The TX and RX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A **SoftwareSerial library** allows for serial communication on any of the Uno’s digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a **Wire library** to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the **SPI library**.

**Programming**

The Arduino Uno can be programmed with the Arduino software (download). Select “Arduino Uno” from the **Tools > Board** menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Arduino Uno comes pre-programmed with a programmer that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the I2C (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use **Atmel’s FLIP software** (Windows) or the **DFU programmer** (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader). See this user-contributed tutorial for more information.

**Automatic (Software) Reset**
Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload. This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It’s labeled “RESET-EN”. You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line; see this forum thread for details.

**USB Overcurrent Protection**

The Arduino Uno has a resettable polyfuse that protects your computer’s USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

**Physical Characteristics**

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16”), not an even multiple of the 100 mil spacing of the other pins.
Lampiran 2 : Datasheet PZEM-004T

AC digital display Multifunction Meter

Product Type: PZEM-004(V3.0)

A. Function
1. Electrical parameter measurement function (voltage, current, active power, energy).
2. Overload alarm function (over power alarm threshold the power flash and the buzzer beeping to alarm).
3. Power alarm threshold preset function (can set power alarm threshold).
4. The reset function of energy key.
5. Store data when power off (store the accumulated energy before power off).
6. Bright red digital display function (display voltage, current, active power, energy).
7. Serial communication function (with TTL serial interface itself, can communicate with a variety of terminal through the pin board, read and set the parameters).

B. Front display and key

I. Display Interface
Display interface is formed by four bright red digital tubes, used to display the voltage, current, power, energy parameters.

II. Display Format
1. Power: Test Range: 0 ~ 225V
   Within 0 ~ 100V, the display format is 0.000 ~ 9999;
   Within 10 ~ 225V, the display format is 1000 ~ 22500.
2. Energy: Test Range: 0 ~ 99999 kWh
   Within 0 ~ 100 kWh, the display format is 0.000 ~ 9999;
   Within 10 ~ 100 kWh, the display format is 1000 ~ 9999;
   Within 100 ~ 1000 kWh, the display format is 10000 ~ 99999;
   1000 ~ 99999 kWh and above the display format is 1000 ~ 99999.
3. Voltage: Test Range: 80 ~ 260VAC
   Display Format is 10.0 ~ 225.0.
4. Current: Test Range: 0 ~ 10A
   Display Format is 0.000 ~ 99.99.

III. Key
There is a key on the panel, it can be used to reset energy.
The method of reset energy: Long press the key for 5 seconds until the digital on energy display window flicker, then release the key. Short press the key again, then the energy data is cleared and quit the flickering state, now the reset operation is completed; if long press for 5 seconds again until no longer flicker, it means exit the reset state.

C. Wiring diagram
The wiring of this module is divided into two parts: the voltage and current test input terminal wiring and the serial communication wiring, as shown in Figure 1; according to the actual needs of the clients, with different TTL pin board to achieve communicate with different terminals.

D. Display Interface

The whole meter panel display window is formed by four windows, they are voltage, current, power and energy; the following are brief description of each parameter display:

1. Voltage Display
Measure and display the current power frequency grid voltage.

2. Current display
Measure and display the current test or appliances) current. This is supplementary instruction that the current test value is from the beginning of 10mA, but the module belongs to high power test equipment, if you care about the mA level current testing accuracy, it is not be recommended.

3. Energy display
Measure and display the current accumulative power consumption. There is supplementary instruction that the minimum unit of the energy function is 0.001kWh, which means it begins to accumulate from 1Wh. Relatively speaking, the resolution is rather high, for the low-power (within 100W) load test, you can observe the accumulative power relatively much.

4. Power display
Measure and display the current load power. There is supplementary instruction that the power test value is from the beginning of 0.001kW, which means it begins to test from 1W, but this module belongs to high power test equipment, if you have the requirement of the testing within 1W, it is not be recommended.

E. Serial communication

This module is equipped with TTL serial data communication interface, you can read and set the relevant parameters via the serial port; but if you want to communicate with a device which has USB or RS232 (such as computer), you need to be equipped with different TTL pin board (USB communication needs to be equipped with TTL to USB pin board; RS232 communication needs to be equipped with TTL to RS232 pin board), the specific connection type as shown in Figure 2. In the below table are the communication protocols of this module:

<table>
<thead>
<tr>
<th>NO.</th>
<th>function</th>
<th>Head</th>
<th>Data1-Data5</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>voltage</td>
<td>B0</td>
<td>CO A8 01 01 00 (Computer sends a request to read the voltage value)</td>
<td>1A</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>current</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B1</td>
<td>C0 A8 01 01 00 (Computer sends a request to read the current value)</td>
<td>1B</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A1</td>
<td>00 11 20 00 00 (Meter reply the current value is 17.32A)</td>
<td>D2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Active power</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B2</td>
<td>C0 A8 01 01 00 (Computer sends a request to read the active power value)</td>
<td>1C</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A2</td>
<td>08 98 00 00 00 (Meter reply the active power value is 2200w)</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Read energy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B3</td>
<td>C0 A8 01 01 00 (Computer sends a request to read the energy value)</td>
<td>1D</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A3</td>
<td>01 86 9F 00 00 (Meter reply the energy value is 999999kWh)</td>
<td>C9</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Set the module address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B4</td>
<td>C0 A8 01 01 00 (Computer sends a request to set the address, the address is 192.168.1.1)</td>
<td>1E</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A4</td>
<td>00 00 00 00 00 (Meter reply the address was successfully set)</td>
<td>A4</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Set the power alarm threshold</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>B5</td>
<td>C0 A8 01 01 14 (computer sends a request to set a power alarm threshold)</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A5</td>
<td>00 00 00 00 00 (Meter reply the power alarm threshold was successfully set)</td>
<td>A5</td>
<td></td>
</tr>
</tbody>
</table>

Illustration of the communication protocol example:

1. Set the communication address: 192.168.1.1
   Send command: B4 C0 A8 01 01 00 00 1E
   Reply data: A4 00 00 00 00 00 00 1E

   **Note:** The above example illustrates that setting the communication address as 192.168.1.1 (the user can set their own address based on their preferences and needs), sending commands and replying data automatically are as shown above, the data are expressed in hexadecimal, the last byte of the sending and replying data are 1E and A4 belong to cumulative sum. At sending commands: B4 + C0 + A8 + 01 + 01 + 00 + 2F = 21E (use the hexadecimal addition), the cumulative sum data is 21E. Take the last two bytes 1E to be used the cumulative sum data in sending commands; data in reply: A4 + 00 + 00 + 00 + 00 + 00 = A4 (use the hexadecimal addition); the cumulative sum data is A4, which is the cumulative sum data in reply.

   The explanation of the cumulative sum is now finished, the following parameter examples are the same as this, there is no explanation any more.

2. Set the power alarm threshold: 20 KW
   Send command: B5 C0 A8 01 01 14 33
   Reply data: A5 00 00 00 00 00 A5

   **Note:** 14 in the sending command is the alarm value (14 is a hexadecimal data representation, which converted to decimal is 20). What you should note is the power alarm value of this module is based on KW units, which means the minimum alarm value is 1KW, the maximum value is 22KW.

3. Read the current voltage
   Send command: B0 C0 A8 01 01 00 1A
   Reply data: A0 00 E6 02 00 00 88
**Note:** Reply voltage data is D1D2D3 = 00 E6 02 00 E6 represent the integer-bit of the voltage, 02 represent the decimal of the voltage, the decimal is one digit, converts 00 E6 to decimal is 230; converts 02 to decimal is 2, so the current voltage value is 230.2V.

4. **Read the current current**
Send command: B1 C0 A8 01 01 00 1B
Reply data: A1 00 11 20 00 00 D2

**Note:** Reply current data is D2D3 = 11 20.11 represent the integer-bit of the current, 20 represent the decimal of the current, the current decimal is two digits, converts 11 to decimal is 17; converts 20 to decimal is 32, so the current current value is 17.32 A.

5. **Read the current power**
Send command: B2 C0 A8 01 01 00 1C
Reply data: A2 08 98 00 00 00 42

**Note:** Reply power data is D1D2 = 08 98, converts 08 98 to decimal is 2200, so the current voltage value is 2200W.

6. **Read the energy**
Send command: B3 C0 A8 01 01 00 1D
Reply data: A3 01 86 9F 00 00 00 00

**Note:** Reply energy data is D1D2D3 = 01 86 9F, converts 01 86 9F to decimal is 999999, so the accumulated power is 999999Wh.

F. **Illustration of the communication**
1. Connect hardware according to the wiring diagram in Figure 1 and 2.
2. After connecting the wire, please choose the communication port. This module's upper computer software support communication port COM1, COM3, COM4, you can check through device manager, if it is not the above communication port, you should modify it through port.

G. **Precautions**
1. This module is suitable for indoor, please do not use outdoor.
2. Applied load should not exceed the rated power.
3. Wiring order can’t be wrong.

H. **Specification parameters**
1. Working voltage: 80 ~ 260VAC
2. Test voltage: 80 ~ 260VAC
3. Rated power: 100A/2200W
4. Operating frequency: 45-65Hz
5. Measurement accuracy: 1.0 grade
**SHENZHEN BONE ELECTRONICS CO., LTD**

1.0 FEATURES
- Display Mode: STN, BLUB
- Display Format: 16 Character x 2 Line
- Viewing Direction: 6 O’Clock
- Input Data: 4-Bits or 8-Bits interface available
- Display Font : 5 x 8 Dots
- Power Supply : Single Power Supply (5V±10%)
- Driving Scheme : 1/16Duty,1/5Bias
- BACKLIT (SIDE) : LED (WHITE)

2.0 ABSOLUTE MAXIMUM

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply for logic</td>
<td>Vdd</td>
<td>-0.3</td>
<td>+7.0</td>
<td>V</td>
</tr>
<tr>
<td>Power supply for LCD Drive</td>
<td>Vcc</td>
<td>Vdd-10.0</td>
<td>Vdd+0.3</td>
<td>V</td>
</tr>
<tr>
<td>Input Voltage</td>
<td>VIL</td>
<td>-0.3</td>
<td>Vdd+0.3</td>
<td>V</td>
</tr>
<tr>
<td>Operating Temperature</td>
<td>Tst</td>
<td>0</td>
<td>+50</td>
<td>°C</td>
</tr>
<tr>
<td>Storage Temperature</td>
<td>Tstg</td>
<td>-10</td>
<td>+60</td>
<td>°C</td>
</tr>
</tbody>
</table>

3.0 ELECTRICAL CHARACTERISTICS

(1=25 °C; Vdd=3.0V±10%, unless otherwise specified)

<table>
<thead>
<tr>
<th>Item</th>
<th>Symbol</th>
<th>Min.</th>
<th>Typ.</th>
<th>Max.</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply for Logic</td>
<td>Vdd</td>
<td>4.7</td>
<td>5.0</td>
<td>5.5</td>
<td>V</td>
</tr>
<tr>
<td>Operating Voltage for LCD</td>
<td>VIL/VIH</td>
<td>--</td>
<td>5.0</td>
<td>--</td>
<td>V</td>
</tr>
<tr>
<td>Input High voltage</td>
<td>VIL</td>
<td>2.2</td>
<td>--</td>
<td>Vdd</td>
<td>V</td>
</tr>
<tr>
<td>Input Low voltage</td>
<td>VIL</td>
<td>--</td>
<td>-0.3</td>
<td>0.6</td>
<td>V</td>
</tr>
<tr>
<td>Output high voltage</td>
<td>VIL</td>
<td>2.4</td>
<td>--</td>
<td>--</td>
<td>V</td>
</tr>
<tr>
<td>Output Low voltage</td>
<td>VIL</td>
<td>--</td>
<td>--</td>
<td>0.4</td>
<td>V</td>
</tr>
<tr>
<td>Power supply current</td>
<td>Idd</td>
<td>1.1</td>
<td>--</td>
<td>--</td>
<td>mA</td>
</tr>
</tbody>
</table>

4.0 MECHANICAL PARAMETERS

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCB Dimension</td>
<td>80.0<em>36.0</em>1.6</td>
<td>mm</td>
</tr>
<tr>
<td>View Dimension</td>
<td>69.5*14.5</td>
<td>mm</td>
</tr>
</tbody>
</table>
5.0 PIN ASSIGNMENT

<table>
<thead>
<tr>
<th>No.</th>
<th>Symbol</th>
<th>Level</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VSS</td>
<td>--</td>
<td>0V</td>
</tr>
<tr>
<td>2</td>
<td>Vdd</td>
<td>--</td>
<td>+5V</td>
</tr>
<tr>
<td>3</td>
<td>V0</td>
<td>--</td>
<td>for LCD</td>
</tr>
<tr>
<td>4</td>
<td>RS</td>
<td>H/L</td>
<td>Register Select: H Data Input, L Instruction Input</td>
</tr>
<tr>
<td>5</td>
<td>R/W</td>
<td>H/L</td>
<td>H--Read, L--Write</td>
</tr>
<tr>
<td>6</td>
<td>E</td>
<td>H,H,L</td>
<td>Enable Signal</td>
</tr>
<tr>
<td>7</td>
<td>DB0</td>
<td>H/L</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>DB1</td>
<td>H/L</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>DB2</td>
<td>H/L</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>DB3</td>
<td>H/L</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>DB4</td>
<td>H/L</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>DB5</td>
<td>H/L</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>DB6</td>
<td>H/L</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>DB7</td>
<td>H/L</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>BLA</td>
<td>--</td>
<td>BLACKLIGHT +5V</td>
</tr>
<tr>
<td>16</td>
<td>BLK</td>
<td>--</td>
<td>BLACKLIGHT 0V</td>
</tr>
</tbody>
</table>

6.0 BLOCK DIAGRAM

7.0 POWER SUPPLY BLOCK DIAGRAM
### 8.0 TIMING CHARACTERISTICS

#### Write Mode (Writing data from MPU to ST7066U)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>T&lt;sub&gt;C&lt;/sub&gt;</th>
<th>T&lt;sub&gt;PNW&lt;/sub&gt;</th>
<th>T&lt;sub&gt;R,T&lt;/sub&gt;F</th>
<th>T&lt;sub&gt;AS&lt;/sub&gt;</th>
<th>T&lt;sub&gt;AH&lt;/sub&gt;</th>
<th>T&lt;sub&gt;DSW&lt;/sub&gt;</th>
<th>T&lt;sub&gt;T&lt;/sub&gt;H</th>
</tr>
</thead>
<tbody>
<tr>
<td>T&lt;sub&gt;C&lt;/sub&gt;</td>
<td>Enable Cycle Time, Pin E</td>
<td>1200</td>
<td>-</td>
<td>-</td>
<td>25</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;PNW&lt;/sub&gt;</td>
<td>Enable Pulse Width, Pin E</td>
<td>140</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;R,T&lt;/sub&gt;F</td>
<td>Enable Rise/Fall Time, Pin E</td>
<td>-</td>
<td>-</td>
<td>25</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;AS&lt;/sub&gt;</td>
<td>Address Setup Time, Pins: RS, RW, E</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;AH&lt;/sub&gt;</td>
<td>Address Hold Time, Pins: RS, RW, E</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;DSW&lt;/sub&gt;</td>
<td>Data Setup Time, Pins: D0 - D7</td>
<td>40</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;T&lt;/sub&gt;H</td>
<td>Data Hold Time, Pins: D0 - D7</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
</tbody>
</table>

#### Read Mode (Reading Data from ST7066U to MPU)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>T&lt;sub&gt;C&lt;/sub&gt;</th>
<th>T&lt;sub&gt;PNW&lt;/sub&gt;</th>
<th>T&lt;sub&gt;R,T&lt;/sub&gt;F</th>
<th>T&lt;sub&gt;AS&lt;/sub&gt;</th>
<th>T&lt;sub&gt;AH&lt;/sub&gt;</th>
<th>T&lt;sub&gt;DSR&lt;/sub&gt;</th>
<th>T&lt;sub&gt;T&lt;/sub&gt;H</th>
</tr>
</thead>
<tbody>
<tr>
<td>T&lt;sub&gt;C&lt;/sub&gt;</td>
<td>Enable Cycle Time, Pin E</td>
<td>1200</td>
<td>-</td>
<td>-</td>
<td>25</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;PNW&lt;/sub&gt;</td>
<td>Enable Pulse Width, Pin E</td>
<td>140</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;R,T&lt;/sub&gt;F</td>
<td>Enable Rise/Fall Time, Pin E</td>
<td>-</td>
<td>-</td>
<td>25</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;AS&lt;/sub&gt;</td>
<td>Address Setup Time, Pins: RS, RW, E</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;AH&lt;/sub&gt;</td>
<td>Address Hold Time, Pins: RS, RW, E</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;DSR&lt;/sub&gt;</td>
<td>Data Setup Time, Pins: D0 - D7</td>
<td>-</td>
<td>-</td>
<td>100</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>T&lt;sub&gt;T&lt;/sub&gt;H</td>
<td>Data Hold Time, Pins: D0 - D7</td>
<td>10</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>ns</td>
</tr>
</tbody>
</table>
Lampiran 4 : Datasheet I2C LCD

I2C interface for LCD

Description:
This LCD2004 is a great I2C interface for 2004 20x2 LCD display. With the limited pin resources, your project may be out of resources using normal LCD shield. With this Serial Interface LCD module, you can need 2 lines (I2C) to display the information. If you already have I2C devices in your project, this LCD module can take up no more resources at all. Fantastic for Arduino based projects.

Specification:
- Compatible with 16x2 and 16x4 LCDs
- Default I2C Address – 0x27
- Address selectable - Range 0x20 to 0x27

Board Layout:

I2C Address Setup:
The LCD2004 board utilized the PCF8574 I/O expander. This nifty little chip provides eight bits of parallel I/O addressable by a I2C bus address – 0x20 to 0x27. SainSmart tied all address leads to Vcc, so the LCD2004 board’s I2C address is permanently fixed at hex 27. This is rather limiting since no additional LCD2004s can be added to the bus.

Anyway, you simply address the board and write an eight bit value which is then presented on the output pins of the PCF8574, which, in this case, are connected to the HD44780 based LCD screen.

```cpp
// Arduino Code
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3, POSITIVE);  // Initialize LCD Display at address 0x27

void setup() {  
  // activate LCD module
  lcd.begin (16,2);  // for 16 x 2 LCD module
  lcd.setBacklightHIGH();
}

void loop() {  
  lcd.home();  // set cursor to 0,0
  lcd.print("Hello, world!");
  lcd.setCursor (0,1);  // go to start of 2nd line
  lcd.print(millis());
  delay(1000);
  lcd.setBacklight(LOW);  // Backlight off
  delay(500);
  lcd.setBacklight(HIGH); // Backlight on
  delay(1000);
}  // END

Check for more info at
https://arduino-info.wikispaces.com/LCD-Blue-I2C
```

© Copyright 2017, Mantech Electronics (Pty) Ltd. All rights reserved.
A MOBICON Company.
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <SoftwareSerial.h>
#include <PZEM004T.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);
PZEM004T pzem(2, 3); // RX, TX connect to TX, RX of PZEM
IPAddress ip(192, 168, 1, 1);
const int sw1 = 10;
const int sw2 = 11;
int next = 0;
int back = 0;

void setup()
{
    pzem.setAddress(ip);
    pinMode(sw1, INPUT);
    pinMode(sw2, INPUT);
    
    lcd.begin();
    lcd.print("Hutama Bagus R");
    lcd.setCursor(0, 1);
   lcd.print("NIM : 13160474");
    delay(3000);
    lcd.clear();
}"
void loop()
{
  menu:
    while (1)
    {
      float v = pzem.voltage(ip);
      if (v < 0.0)v = 0.0;
      {
        lcd.setCursor(0, 0);
        lcd.print("V=");
        lcd.setCursor(2, 0);
        lcd.print(v);
      }
      float i = pzem.current(ip);
      if (i < 0.0)i = 0.0;
      {
        lcd.setCursor(0, 1);
        lcd.print("A=");
        lcd.setCursor(2, 1);
        lcd.print(i);
      }
    }
    lcd.setCursor(8,0);
    lcd.print("|");
    lcd.setCursor(8,1);
    lcd.print("|");
}
float p = pzem.power(ip);
if (p < 0.0)p = 0.0;
{
  lcd.setCursor(9, 0);
  lcd.print("W=");
  lcd.setCursor(11, 0);
  lcd.print(p);
  lcd.print(p);
}

float e = pzem.energy(ip);
if (e >= 0.0)e = 0.0;
{
  lcd.setCursor(9, 1);
  lcd.print("Wh= ");
  lcd.setCursor(12, 1);
  lcd.print(e);
  lcd.print(e);
}

next = digitalRead(sw1);
back = digitalRead(sw2);
if (next == LOW) {delay(100);lcd.clear();goto menu1;}
if (back == LOW) { }
}

menu1:
  while (1)
  {
    float v = pzem.voltage(ip);
    if (v < 0.0)v = 0.0;
float i = pzem.current(ip);
if (i < 0.0)i = 0.0;

float p = pzem.power(ip);
if (p < 0.0)p = 0.0;

lcd.setCursor(0,0);
lcd.print("Bocor: ");
lcd.setCursor(6,0);
lcd.print((p)/1000);
lcd.setCursor(13,0);
lcd.print("Kwh");

float r = ((p)/1000) * 1467 * 24* 30;
lcd.setCursor(0, 1);
lcd.print("Rugi: " );
lcd.setCursor(5, 1);
lcd.print("Rp.");
lcd.setCursor(8, 1);
lcd.print(r);

delay(500);

next = digitalRead(sw1);
back = digitalRead(sw2);
if (next == LOW) { }
if (back == LOW) {delay(100);lcd.clear();goto menu;}
} 
}
Lampiran 6 : Skema Rangkaian Keseluruhan
<table>
<thead>
<tr>
<th>No</th>
<th>Nama Komponen</th>
<th>Harga Komponen</th>
<th>Gambar Komponen</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Arduino Uno R3 DIP + USB Cable</td>
<td>Rp.85.000</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>PZEM-004T Sensor Arus AC TTL Serial</td>
<td>Rp.142.000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>LCD 16x2 + I2C Serial Backpack</td>
<td>Rp. 28.900</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Tact Tactile Push Button</td>
<td>Rp.500</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Kabel Jumper Arduino (Male to Male) 20 Pcs</td>
<td>Rp.8.500</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Kabel Jumper Arduino (Male to Female) 20 Pcs</td>
<td>Rp.8.500</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>PCB Kosong Berlubang</td>
<td>Rp.7.000</td>
<td></td>
</tr>
</tbody>
</table>
Lampiran 8 : Foto Alat
Lampiran 9: Tarif Listrik PLN 2019 (\textit{Tariff Adjustment})


tbl

<table>
<thead>
<tr>
<th>NO.</th>
<th>GOL. TARIF</th>
<th>BATAS DAYA</th>
<th>REGULER</th>
<th>PRA BAYAR (Rp/kWh)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>R-1/TR</td>
<td>1.300 VA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>2.</td>
<td>R-1/TR</td>
<td>2.200 VA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>3.</td>
<td>R-2/TR</td>
<td>3.500 VA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>4.</td>
<td>R-3/TR</td>
<td>6.000 VA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>5.</td>
<td>B-2/TR</td>
<td>6.600 VA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>6.</td>
<td>B-3/TM</td>
<td>200 kVA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>7.</td>
<td>I-3/TM</td>
<td>4.000 kVA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>8.</td>
<td>I-4/TT</td>
<td>30.000 kVA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>9.</td>
<td>P-1/TR</td>
<td>6.600 VA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>10.</td>
<td>P-2/TM</td>
<td>2.000 kVA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>11.</td>
<td>P-3/TT</td>
<td>6.600 VA</td>
<td>1.467,28</td>
<td>1.467,28</td>
</tr>
<tr>
<td>12.</td>
<td>UT/TT</td>
<td>1.644,52</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Catatan:

1) Diterapkan Rekening Minimum (RM):
   - RM1 = 40 (Jam Nyala) x Daya tersambung (kVA) x Biaya Pemakaian.
   - RM2 = 40 (Jam Nyala) x Daya tersambung (kVA) x Biaya Pemakaian LWBP.

2) Diterapkan Rekening Minimum (RM):
   - RM3 = 40 (Jam Nyala) x Daya tersambung (kVA) x Biaya Pemakaian WBP dan LWBP.

3) Biaya kelebihan pemakaian daya reaktif (kVAh) dikenakan dalam hal faktor daya rata-rata setiap bulan kurang dari 0,8 (delapan puluh lima persen) per seratus.

K : Faktor perbandingan antara harga WBP dan LWBP sesuai dengan karakteristik beban sistem kelistikan selama (1,4 ≤ K ≤ 2), ditetapkan oleh Direksi Perusahaan Perseroan (Persero) PT Perusahaan Listrik Negara.

WBP : Waktu Beban Puncak.
LWBP : Luar Waktu Beban Puncak.