

Study on embedded system for voltage and current measurement using microcontroller and sensors

Aakash N. Patel[a], Dr. K. G. Raval[b], Dr. V. G. Joshi[c]

[a]Department of Physics,VNSGU Surat−395007,India
[b]Narmada college of Science & Commerce,Zadeshwar,Bharuch−392011,India
[c]Department of Physics,VNSGU Surat−395007,India

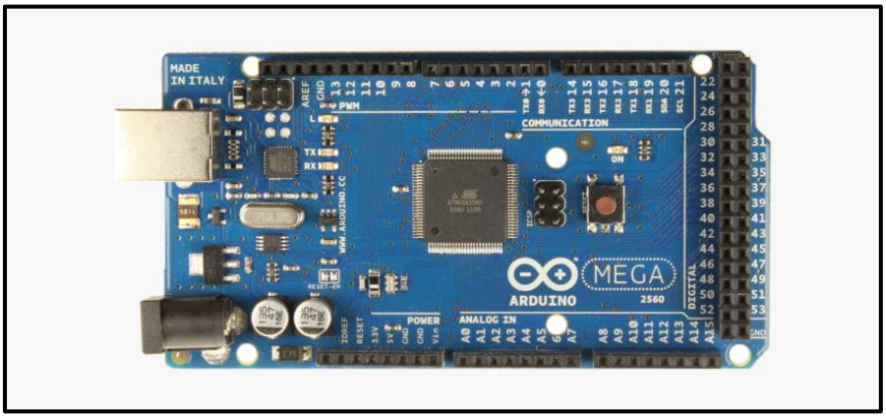
Abstract

Microcontroller is widely used in different sophisticated instruments as they are highly reliable and easily programmable as per our need .Real time monitoring of voltage and current has very wide range of applications. In this paper an experimental system based on Atmega 2560 microcontroller was developed. The main components of the system were Atmega 2560 microcontroller, voltage sensor and current sensor. In this paper data acquisition process between microcontroller and computer is also introduced. The method used for transferring data is serial communication.
Keywords: Microcontroller, Voltage sensor, Sensor interface

Introduction

Microcontroller is very important part of any embedded system as it is required to interpret variation in different voltages of sensor. Microcontroller consists of various input and output port. In this paper the method to design and coding of embedded system is discussed .This embedded system can measure different variation in voltage and current in real time. This embedded system can also measure voltage for more than one input simultaneously. This embedded system consists of four different part development board, voltage sensor, current sensor and data acquisition software.

1. Development board



Fig(1) ATmega 2560 Microcontroller

Development board will be main component for this system. It will retrieve data from the sensors then process it and send it to the serial communication port .In this sensor module; development board based on Atmega 2560 microcontroller is used. It has 15 analog inputs, 13 PWM output 5 communication port and 31 digital input/output.[2]

Arduino is an open source, computer hardware and software company, project, and user community that designs and manufactures microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces. The Arduino company provides integrated development environment to load sketch on Arduino board.

 https://doi.org/10.5281/zenodo.14551505

The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, 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 Mega 2560 board is compatible with most shields designed for the Uno and the former boards.

The Mega 2560 board can be programmed with the Arduino Software (IDE). The ATmega2560 on the Mega 2560 comes preprogramed with a boot loader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol. You can also bypass the boot loader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP.

The Mega 2560 has a resettable polyfuse that protects 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.

The Mega 2560 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-wart) or battery. The adapter can be connected by plugging a 2.1mm centre-positive 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 five volts and the board may become unstable. 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:

1) **Vin**- The input voltage to the board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
2) **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 advise it.
3) **3V3 A**- 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
4) **GND-**. Ground pins.
5) **IOREF**- This pin on the board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

The ATmega2560 has 256 KB of flash memory for storing code (of which 8 KB is used for the bootloader), 8 KB of SRAM and 4 KB of EEPROM (which can be read and written with the EEPROM library).

Each of the 54 digital pins on the Mega 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 20 mA as recommended operating condition and has an internal pull-up resistor (disconnected by default) of 20-50 k ohm. A maximum of 40mA is the value that must not be exceeded to avoid permanent damage to the microcontroller.

The Mega 2560 has 16 analog inputs, 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 analogReference() function.

Communication

The Mega 2560 board has a number of facilities for communicating with a computer, another board, or other microcontrollers. The ATmega2560 provides four hardware UARTs for TTL (5V) serial communication. An ATmega16U2 (ATmega 8U2 on the revision 1 and revision 2 boards) on the board channels one of these over USB and provides a virtual com port to software on the computer (Windows machines will need a .inf file, but OSX and Linux machines will recognize the board as a COM port automatically. The Arduino Software (IDE) includes a serial monitor which allows simple textual data to be sent to and from the board. The RX and TX LEDs on the board will flash when data is being transmitted via the ATmega8U2/ATmega16U2 chip and USB connection to the computer (but not for serial communication on pins 0 and 1). A Software Serial library allows for serial communication on any of the Mega 2560's digital pins.

https://doi.org/10.5281/zenodo.14551505

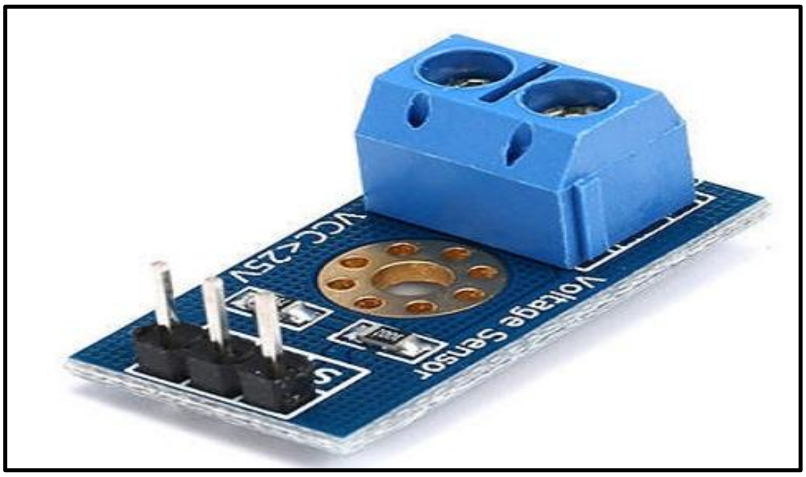
Physical Characteristics and Shield Compatibility

The maximum length and width of the Mega 2560 PCB are 4 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Three screw holes allow the board to be attached to a surface or case.
The Mega 2560 is designed to be compatible with most shields designed for the Uno and the older Arduino boards. Digital pins 0 to 13 (and the adjacent AREF and GND pins), analog inputs 0 to 5, the power header, and ICSP header are all in equivalent locations. Furthermore, the main UART (serial port) is located on the same pins (0 and 1), as are external interrupts 0 and 1 (pins 2 and 3 respectively). SPI is available through the ICSP header on the mega 2560

Automatic (Software) Reset

Rather then requiring a physical press of the reset button before an upload, the Mega 2560 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 is connected to the reset line of the ATmega2560 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 (IDE) 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 Mega 2560 board 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 ATMega2560. 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 Mega 2560 board 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.

2. Voltage sensor



Fig(2) Voltage Sensor

This sensor is used to monitor, calculate and determine the voltage supply. This sensor can determine the AC or DC voltage level. The input of this sensor can be the voltage whereas the output is the switches, analog voltage signal, a current signal, an audible signal, etc. Some sensors provide sine waveforms or pulse waveforms like output & others can generate outputs like AM (Amplitude Modulation), PWM (Pulse Width Modulation) or FM (Frequency Modulation). The measurement of these sensors can depend on the voltage divider. This sensor includes input and output. The input side mainly includes two pins namely positive and negative pins. The two pins of the device can be connected to the positive & negative pins of the sensor. The device positive & negative pins can be connected to the positive & negative pins of the sensor. The output of this sensor mainly includes supply voltage (Vcc), ground (GND), analog output data. [1]

https://doi.org/10.5281/zenodo.14551505

Embedded C code for voltage measurement

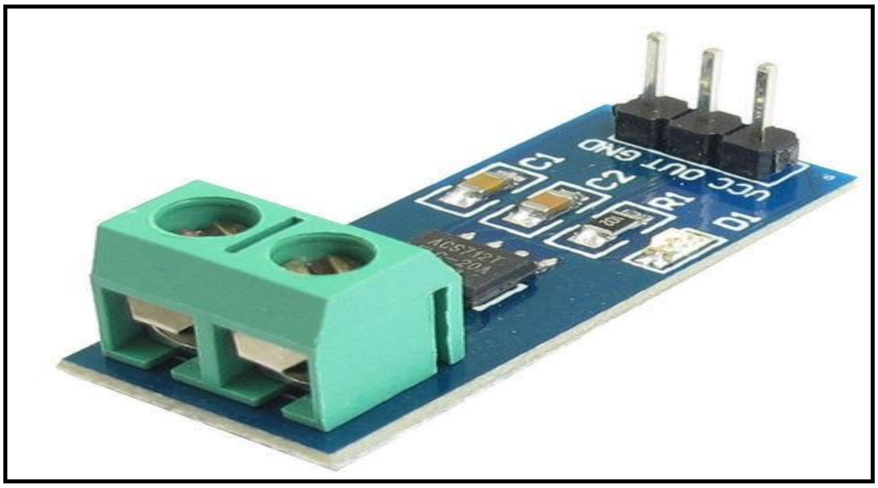
```
int sensorValue0 = 0;
int sensorValue1 = 0;
int sensorValue2 = 0;
int sensorValue3 = 0;
float voltage0 = 0;
float voltage1 = 0;
float voltage2 = 0;
float voltage3 = 0;
void setup() {
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT);
  pinMode(A3,INPUT);
  Serial.begin(9600);
  // initialize serial communication at 9600 bits per second:
  Serial.println("LABEL,V1,V2,V3,V4");
  // labeling column of output data
}
// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0 to 3:
  int sensorValue0 = analogRead(A0);
  int sensorValue1 = analogRead(A1);
  int sensorValue2 = analogRead(A2);
  int sensorValue3 = analogRead(A3);
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 25V):
  float voltage0 = sensorValue0 * (25.0 / 1023.0);
  float voltage1 = sensorValue1 * (25.0 / 1023.0);
  float voltage2 = sensorValue2 * (25.0 / 1023.0);
  float voltage3 = sensorValue3 * (25.0 / 1023.0);
  // print out the value you read:
  Serial.println(voltage0);
  Serial.println(",");
  Serial.println(voltage1);
  Serial.println(",");
  Serial.println(voltage2);
  Serial.println(",");
  Serial.println(voltage3);
  Serial.println(",");
}
```

Description of embedded C code

In this code four different voltage sensor is connected at analog pins of microcontroller and then variation of analog voltage is measured from 0 to 1023 which is resolution of analog voltage detected by the microcontroller. then the measured value between 0 to 1023 is further calibrated between 0 to 25 voltage which is range of voltage sensor. After the calibration process is executed the value is further transferred to serial communication port which can be captured by data acquisition software.

 https://doi.org/10.5281/zenodo.14551505

3. Current sensor



Fig(3) Current Sensor

Current Sensor detects the current in a wire or conductor and generates a signal proportional to the detected current either in the form of analog voltage or digital output. Current Sensing is done in two ways – Direct sensing and Indirect Sensing. In Direct sensing, to detect current, Ohm's law is used to measure the voltage drop occurred in a wire when current flows through it.

A current-carrying conductor also gives rise to a magnetic field in its surrounding. In Indirect Sensing, the current is measured by calculating this magnetic field by applying either Faraday's law or Ampere law. Here either a Transformer or Hall effect sensor or fiber optic current sensor are used to sense the magnetic field.

ACS712 Current Sensor uses Indirect Sensing method to calculate the current. To sense current , low-offset Hall sensor circuit is used in this IC. This sensor is located at the surface of the IC on a copper conduction path. When current flows through this copper conduction path it generates a magnetic field which is sensed by the Hall effect sensor. A voltage proportional to the sensed magnetic field is generated by the Hall sensor, which is used to measure current.The proximity of the magnetic signal to the Hall sensor decides the accuracy of the device. Nearer the magnetic signal higher the accuracy. ACS712 Current Sensor is available as a small, surface mount SOIC8 package. In this IC current flows from Pin-1 and Pin-2 to Pin-3 and Pin-4. This forms the conduction path where the current is sensed. Implementation of this IC is very easy.

ACS712 can be used in applications requiring electrical isolation as the terminals of the conduction path are electrically isolated from the IC leads. Thus, this IC doesn't require any other isolation techniques. This IC requires a supply voltage of 5V. Its output voltage is proportional to AC or DC current. ACS712 has a nearly zero magnetic hysteresis. **ACS712ELCTR-05B-T** can measure 5 to -5 Ampere current. Where 185mV change in Output voltage from initial state represents 1-Ampere change in Input current.

Embedded C code for Current measurement

```
void setup() {
  Serial.begin(9600);
 //Start Serial Monitor to display current read value on Serial monitor
}
void loop() {
int x=0;
float AcsValue=0.0;
float Samples=0.0;
float AvgAcs=0.0;
float AcsValueF=0.0;
  for (int x = 0; x < 150; x++){
//Get 150 samples
  AcsValue = analogRead(A0);
    //Read current sensor values
  Samples = Samples + AcsValue;
//Add samples together
```

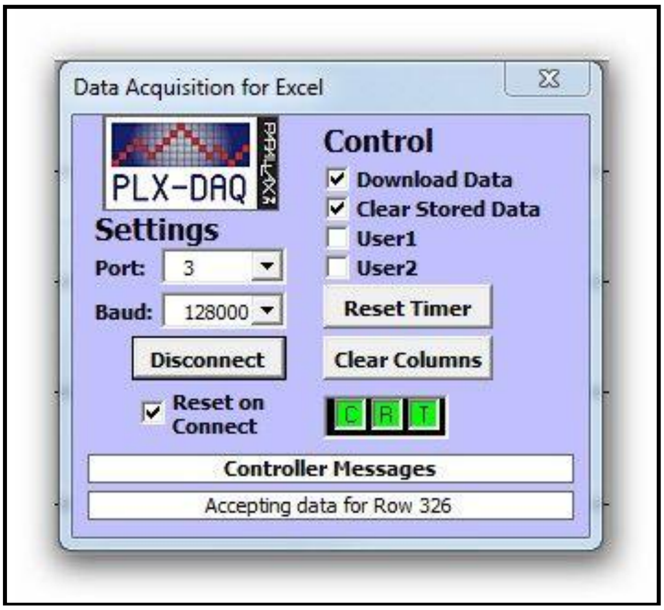
 https://doi.org/10.5281/zenodo.14551505


```
  delay (3);
 // let ADC settle before next sample 3ms
}
AvgAcs=Samples/150.0;
//Taking Average of Samples
//((AvgAcs * (5.0 / 1024.0)) is converitng the read voltage in 0-5 volts
//2.5 is offset( arduino is working on 5v so the viout at no current comes
//out to be 2.5 which is out offset. If  arduino is working on different voltage than
//we must change the offset according to the input voltage)
//0.185v(185mV) is rise in output voltage when 1A current flows at input
AcsValueF = (2.5 - (AvgAcs * (5.0 / 1024.0)) )/0.185;
Serial.print(AcsValueF);//Print the read current on Serial monitor
delay(50);
}
```

Description of embedded C code

In this code four current sensor is connected at analog pins of microcontroller and then variation of analog voltage is measured from 0 to 1023 which is resolution of analog voltage detected by the microcontroller. then the measured value between 0 to 1023 is further calibrated between 0 to 5 Ampere which is range of current sensor .There is also requirement of offset during calibration which is included in code. There is 0.185 volt change for 1 amp current flow so we must include this during converting voltage to current. After the calibration process is executed the value is further transferred to serial communication port which can be captured by data acquisition software.

4. Data acquisition software[3]



Fig(5) PLX-DAQ Interface

Arduino board has one usb port from which we can communicate by serial communication or we can use tx and rx pin for serial communication with microcontroller. We are in need to get data instantly as it generates by microcontroller so we use serial port for real time data acquisition. Arduino IDE software comes up with it's inbuilt serial monitor and serial plotter but it doesn't helpful for this application.

PLX-DAQ is a Parallax microcontroller data acquisition add-on tool for Microsoft Excel. Any of our microcontrollers connected to any sensor and the serial port of a PC can now send data directly into Excel. PLX-DAQ has the following features:

- It can Record up to 26 columns of data.
- It can Mark data with real-time (hh:mm:ss) or seconds since reset.
- It can communicate with microcontroller on Baud rates up to 128K.
- It can Supports up to 15 communication port.

 https://doi.org/10.5281/zenodo.14551505

To get the data from Arduino You only need to download and install it, it should work fine. After installation, it will automatically create a folder named PLX-DAQ on your Desktop in which you will find a shortcut named PLX-DAQ Spreadsheet. When you want to use your Arduino to send data to excel, just open up the shortcut. Now by following further procedure one can get data from Arduino by this software.

- open the shortcut to your PLX-DAQ Spreadsheet
- excel will say "This application is about to initialize ActiveX…", just click OK
- a new window named Data Acquisition for Excel will appear
- select the usb port your Arduino is connected to (if it doesn't work at first, go through the list of ports)where it says Baud, just select the number you put in your code at Serial.begin()
- create an empty graph
- select which columns of data you want on the graph for the x and y axis
- click collect data on PLX-DAX and it should start collecting the data

Conclusion

This research has to do with the measurement of electronic quantities such as voltage and current through microcontroller and sensor. The built system works perfectly and it also provide data in real time. Measurements of electrical quantities serve to set the condition in the sketch-program code written in embedded C within the Arduino IDE environment.This paper will serve as a good basis for further research in real time data analysis for solar rooftop or similar type of multiple voltage source devices.

References

[1] Mustafa, Kujtim & Mustafa, Ragmi & Ramadani, Refik. (2023). Measuring the Voltage, Current and Resistance of the LDR Sensor through the Arduino UNO. Asian Journal of Research in Computer Science. 16. 211-222. 10.9734/ajrcos/2023/v16i4383.

[2] Aakash N. Patel, & Dr. K. G. Raval. (2024). Ultrasonic Three Dimensional Scanner. Periodico di Mineralogia, Volume 93, No. 5 ,https://doi.org/10.5281/zenodo.13822555

 https://doi.org/10.5281/zenodo.14551505