VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Mingliang Zhao

DATA ACQUISITION SYSTEM USING ATMEGA128L

Information Technology

2012

FORWORD

This thesis was made in Vaasa University of Applied Science during the academic year of 2011-2012.

The special thank goes to my helpful supervisor, Mr. Menani Smail. The supervision and support that he gave truly help the progression and smoothness of the project. The co-operation is much indeed appreciated.

My grateful thank also goes to the Mr. Martila Jukka. He offered me the thesis topic and gave me technique support during the project design.

Mingliang Zhao

01, Feb, 2012

VAASAN AMMATTIKORKEAKOULU
UNIVERSITY OF APPLIED SCIENCES
Degree Program of Electronics and Information Technology

Author	Mingliang Zhao
Title	DATA ACQUISITION SYSTEM USING ATMEGA128L
Year	2012
Language	English
Pages	52+ 2 Appendices
Name of Supervisor	Menani Smail

Data Acquisition system is developed to monitor the environmental temperature and control the microcontroller. System consists of remote terminal unit (RTU) and supervisory station (PC). Remote terminal unit connects to the physical equipment sensor PT-100, which comprises a signal conditioning circuit and a microcontroller. The signal conditional circuit transforms the physical temperature parameters to analog signals and manipulates these analog signals. Then, microcontroller will convert analog to digital signals and send them through RS232 cabling. The term "supervisory station", also can be called as master station, refers to the application (GUI) responsible for communicating with the RTUs which are distributed to measure the temperature of different places. In this simple DATA ACQUISITION system, the supervisory station is only composed of a single PC. The thesis of DATA ACQUISITION system mainly concentrates on the electronic and software design for this remote terminal unit, as well as the software development for the GUI on master station.

Keywords: RTD, Atmega128, MySQL, PADS, MODBUS, Qt

CONTENTS

LIST OF FIGURES AND TABLES

LIST OF APPENDICES

1 INTRODUCTION

Control is not only the information processing, it also implies the direct interaction with the physical world. Control systems include sensors and actuators, which is needed to ensure that our automation system can help us manage our activities and environments in desired ways. Sensors provide inputs to control and automation systems by extracting information from the physical world.

Recently, Data Acquisition system is widely used in the world. With the development of the modern industry, process variables of temperature, pressure, flow rate and liquid level are main objects of observation. Temperature is one of most important process variables among them. For example, in the fields of metallurgical industry, chemical industry, power industry, machinery processing industry, food industry and so on, the temperature control system are required for all kinds of furnaces, reactors or boilers

Data Acquisition system is aimed to monitor the environment temperature and control the microcontroller. The design of the system can also help us to learn the protocol MODBUS which is widely used in electronic devices.

The objective of doing this project is to acquire knowledge and understandings of the control system. Apart from this, the programming language and communication method will be learned and mastered.

In this thesis, the chapter 2 raises up the problems needed to be considered according to the objective of the project.

The chapter 3 introduces the system structure and how the system works.

The chapter 4 finds out the method to solve the problems in chapter 2.

The chapter 5 introduces the design and implementation according to the structure of system.

The chapter 6 gives way of testing the system.

The chapter 7 discusses the frequently asked questions and the problems that have not been solved.

The chapter 8 summarizes the process of doing this project.

2 REQUIREMENTS

2.1 Objective

In the Data Acquisition system, Remote Terminal Unit (RTU) will measure the environmental temperature data between -60 - 120 ℃. The GUI application on master station (PC) will send the requests of getting the temperature data or setting the status of ADC on microcontroller. In addition, the GUI application will show the frames of different requests and feedbacks. Every frame, which is a pack of transmit/receive data, represents the different function in MODBUS protocol. The temperature data got from the RTU will finally be saved to the Database for further processing.

2.2 Problems to be solved

If the objective above were fulfilled, the following problems would be solved:

1. To measure the temperature is actually to transform physical parameters to analog signals. So different sensors should be compared and chosen according to their own attributes. In the control system, the analog signal always needs to be manipulated for further processing. In this case, the signal conditioning circuit should be designed for the sensor.

2. In computer architecture, digital signal is commonly used in microcontroller or computers. So the digital signal processing is necessary for transforming the analog signals.

3. To bridge the master station and RTU, communication device is necessary to be considered and communicational protocol also needs to be taken into account according to the devices we choose.

4. On the master station, user interface application is used to control the sensor system. So the following functions should be integrated at least in the application according to the objective:

1) Display the temperature data, sensor address and time at least.
2) Turn on/off the sensor system.
3) Display the data frame.
4) Analyze the protocol.

5. For further processing the temperature data or massive data storage, the Database is needed. The system should have some functions to save the temperature data to the Database. For example, if people want to read the temperature record through internet, they can just connect their own applications to the Database.

3 PROCESS DESCRIPTION

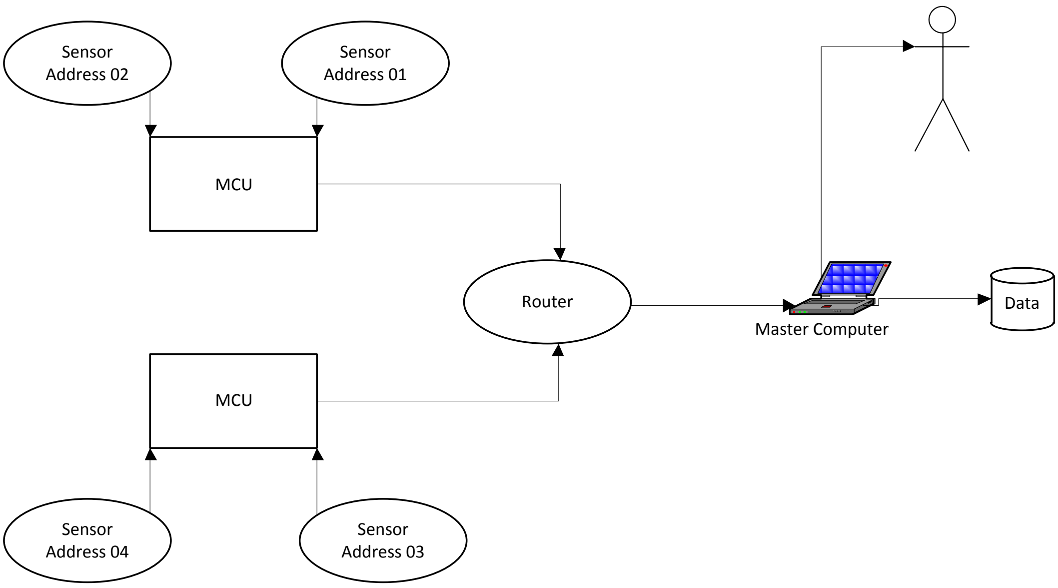


Figure 1 System Structure

In Figure 1, System is divided into two main parts. They are the remote terminal unit (RTU) and the master station respectively. The RTU is also composed of signal conditioning PCB part and microcontroller part.

1. The part of signal conditioning PCB is shown where sensor address is marked. The PCB connects to the temperature sensor PT-100 which will transform the physical temperature parameters to analog signals. Then the signals will be amplified by the signal conditioning circuit on PCB. In this case, the processed analog signals can be output to the microcontroller.

2. The part of microcontroller is shown where the MCU is marked. The microcontroller will digitize the input analog signals with ADC and communicates with master station through RS232 or RS385 interface. Moreover, the MODBUS for RTU protocol is utilized in the communication between microcontroller and master station.

3. The part of master station is shown where Master Computer and Data is marked. The GUI application is designed on the master station, which monitors the messages of communication and displays the temperature da-

ta. Furthermore, any displayed temperature data can be saved to the database. It should be noticed that all the frames in the communication belong to the protocol MODBUS and the Database mySQL is located in the school (VAMK) site.

4 ANALYSIS

4.1 Temperature measurement device

4.1.1 Sensor camparison

There are two kinds of sensors which are used widely. They are thermocouple and resistance temperature detector (RTD) respectively.

A thermocouple is a device consisting of two different conductors (usually metal alloys) that produce a voltage, proportional to a temperature difference, between either ends of the two conductors.

RTD is sensor used to measure temperature by correlating the resistance of the RTD element with temperature.

Feature	**Thermocouple**	**RTD**
Response time	Better	
Maximum temperature	Higher	
Ruggedness	Better	
Cost efficiency	Better	
Accuracy		Better
Long-term stability		Better
Standardization		Better

Table 1 Comparison of temperature sensors' attributes: /1/

Table 1 compares the attributes of two different sensors. It shows the advantage for each sensor.

RTDs are generally considered to be among the most accurate temperature sensors available. In addition to offering very good accuracy, they provide excellent stability and repeatability. RTDs also feature high immunity to electrical noise and are, therefore, well suited for applications in process and industrial automation environments, especially around motors, generators and other high voltage equipment.

Since thermocouples measure wide temperature ranges and are relatively rugged, they are very often used industrial and process applications where accuracy may be a less important factor.

However, the data acquisition system is designed to measure the room temperature which is in a small range from -60 - 120 ℃. The ideal temperature accuracy is 0.1 ℃. So according to the previous statement, the RTD is more suitable for our design.

Sensor type PT-100 belongs to the resistance temperature detector (RTD). The measurement range is -200 to 850 ℃.

It should be noticed that since the heat conduction in resistor takes time, the response time of thermocouple is better than RTD. It means that the sensor PT-100 cannot reflect the environment temperature instantly.

There is a relationship between temperature and corresponding resistance. For almost all pure-metal resistors, the relationship is in proportional. According to the experimental statistic, the relationship of PT-100 is in formula as below:

$$R_{\mathrm{T}} = 0.385 * \Delta \mathrm{T} + R_{T0}$$

R_T is the resistance at temperature T.

R_{T_0} is the resistance at temperature T_0.

ΔT is the difference of T and T_0.

4.1.2 Signal conditioning circuit

In the signal conditioning circuit, there are two important modules. The first module generates the analog signals according to the physical temperature parameters.

The second module would amplify the analog signals for further use. Hence, in the first module, the way of the RTD or PT-100 being connected to the signal conditioning circuit is very important.

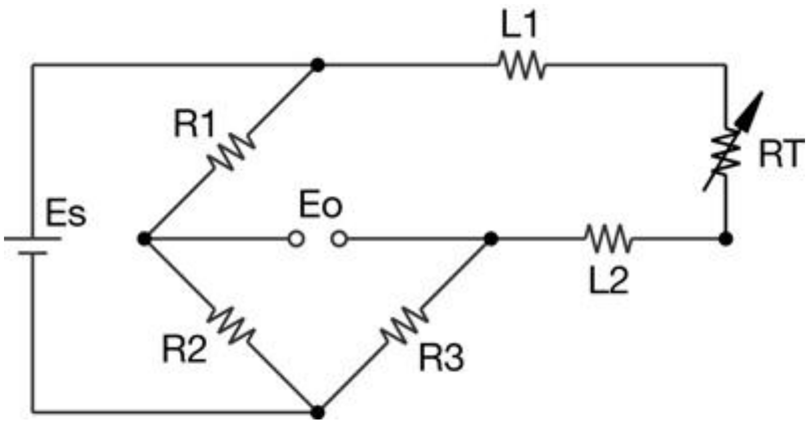


Figure 2 2-wire circuit: /1/.

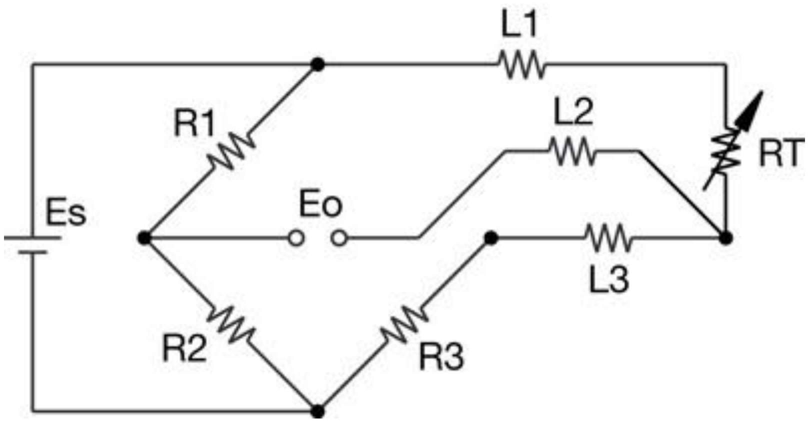


Figure 3 3-wire circuit: /1/.

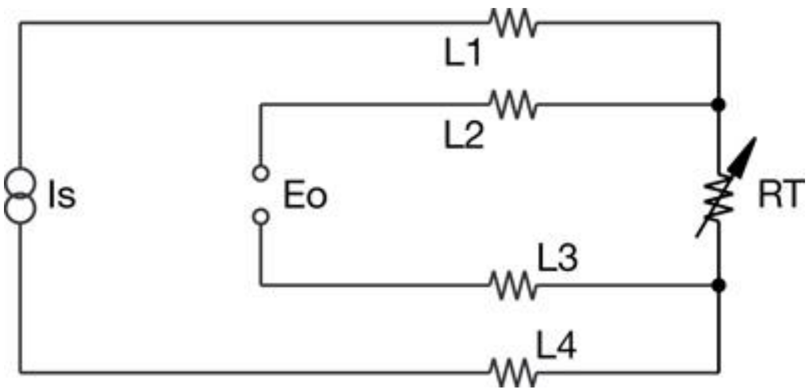


Figure 4 4-wire circuit: /1/.

Es is the supply voltage;

Eo is the output voltage;

R1, R2, and R3 are fixed resistors;

L1 and L2 are lead resistance.

RT is the RTD.

PT-100 connection has 3 common ways to be implemented to the circuit.

Figure 2 shows that the temperature is determined by measuring resistance. But when measuring the resistance of the sensing element (RT), the resistances of the leads (L1 and L2) and cables (R1, R2, and R3) are also measured. So 2-wire connection gives an error!

Figure 3 shows the way to compensate the error with 3-wire connection. In this circuit there are three leads coming from the RTD instead of two. L1 and L3 carry the measuring current while L2 acts only as a potential lead. No current flows through it while the bridge is in balance. Since L1 and L3 are in separate arms of the bridge, resistance is canceled.

Figure 4 shows another way of choice to do the compensation with 4-wire connection. 4-wire RTD circuits not only cancel lead wires but remove the effects of mismatched resistances such as contact points. A common version is the constant current circuit shown here. Is drives a precise measuring current through L1 and L4; L2 and L3 measure the voltage drop across the RTD element. Eo must have high impedance to prevent current flow in the potential leads.

4.2 Analog to Digital Converter resolution

Analog to Digital Converter (ADC) is an electronic device that converts voltage to digital number proportional to the magnitude of the voltage which we call it as reference voltage.

The resolution of the converter indicates the number of discrete values it can produce over the range of analog values.

The configuration of ADC resolution in microcontroller has normally two types. They are respectively 8-bit and 10-bit. Here the n-bit means there are 2^n quantization levels. In practice, Quantization error and non-linearity are intrinsic to any analog-to-digital conversion. However, through the resolution selection, the error can be reduced.

In the signal conditioning circuit PCB, the output voltage range is between 0.7-1.5V which refers to the temperature approximately between -60-120℃. If the system requires the accuracy to 1℃, the quantization levels should be around 180 for ADC. In the 8-bit ADC, the quantization levels are (1.5-0.7)/5*2^8= 40. But, in the 10-bit ADC, the quantization levels are (1.5-0.7)/5*2^10=163. Apparently, the 10-bit is approaching to the required value 180. So in order to reduce the quantization error, the better choice is 10-bit resolution for the ADC.

4.3 The STarter Kit (STK)

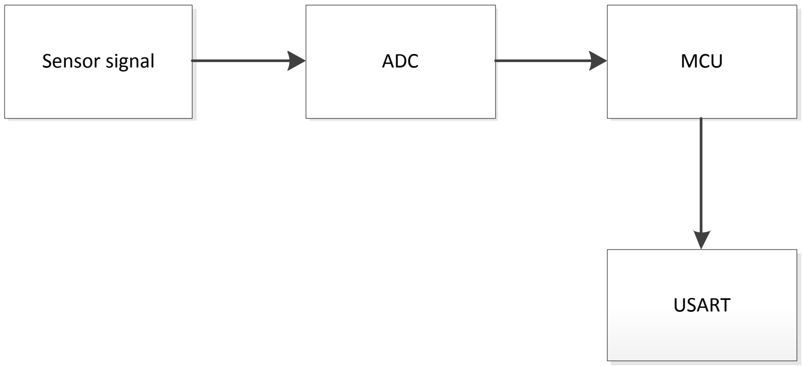
The Starter Kit is a collection of commonly used components for microcontroller. It can help us initialize the project when the specific components are not determined to use in the project.

The STK501 board is a top module designed to add Atmega128L support to STK500 development board. It consists of microprocessor Atmega128L, the 8-bit and 16-bit timer/counter, 10-bit analog to digital converter (ADC), The Universal Synchronous and Asynchronous serial Receiver and Transmitter (USART) and the memory.

J-Tag is a programming used device which can be connected to the J-Tag interface on the board.

AVR-studio is the programming IDE that uses gcc-compiler.

4.3.1 Data flow in microcontroller



Atemega128 microcontroller receives input signals from the signal conditioning circuit. The signal will be processed by the analog to digital converter (ADC) firstly. Then the output digital signal from the ADC will be sent to the temperate memory for MCU. If the MCU need communicate with master station, the data will be transmitted through the RS232 or RS385 interface using the USART.

4.4 Protocol

A communications protocol is a system of digital message formats and rules for exchanging those messages in or between computing systems and in telecommu-nications.

Modbus is a serial communications protocol which is widely used in industrial electronic devices nowadays.

The reasons to use the Modbus in industrial environment are:

1. It has been developed with industrial applications in mind
2. It is openly published and royalty-free
3. It is easy to deploy and maintain
4. It moves raw bits or words without placing many restrictions on vendors

Modbus is often used to connect a supervisory computer with a remote terminal unit (RTU) in Data Acquisition systems.

4.5 User Interface

To fulfill the requirements of the system, the following functions need to be realized at least:

1. Supervise the communication frames in MODBUS.
2. Display received temperature data.
3. Display formatted temperature data.
4. Command to get the temperature data from the sensor.
5. Command to save temperature data to Database.

In order to be more convenient for the users, graphic user interface is proposed to be built. For this reason, Qt language which is designed for GUI application is chosen.

Qt language allows you to write advanced applications and UIs once, and deploy them across desktop and embedded operating systems without rewriting the source code saving time and development cost.

Qt Creator is a cross-platform integrated development environment (IDE) tailored to the needs of Qt developers. It provides the code editor and integrated UI designer. The UI designer offer simple tools for the programming.

4.6 Database

Database can be used to save the data permanently while the previous GUI application can only display the data temporarily. So it is necessary to save the data which will be used in future.

MySQL is the most popular open source database server in existence currently. It is commonly used for Web applications, embedded applications and has become a popular alternative to proprietary database systems because of its speed and reliability.

It should be noticed that MySQL can run on UNIX, Windows and Mac OS and the system connects the Database located in VAMK through the internet.

5 DESIGN AND IMPLEMENTATION

5.1 Signal conditioning circuit design

The objective of this circuit is generating appropriate electronic signals according to the temperature physical parameters. The circuit is divided into two modules as its functions. The first module generates constant current through PT-100 to produce the voltage signal for the PT-100 four-wire connection requires the constant current input. The second module amplifies the signal to make it available for ADC in microcontroller.

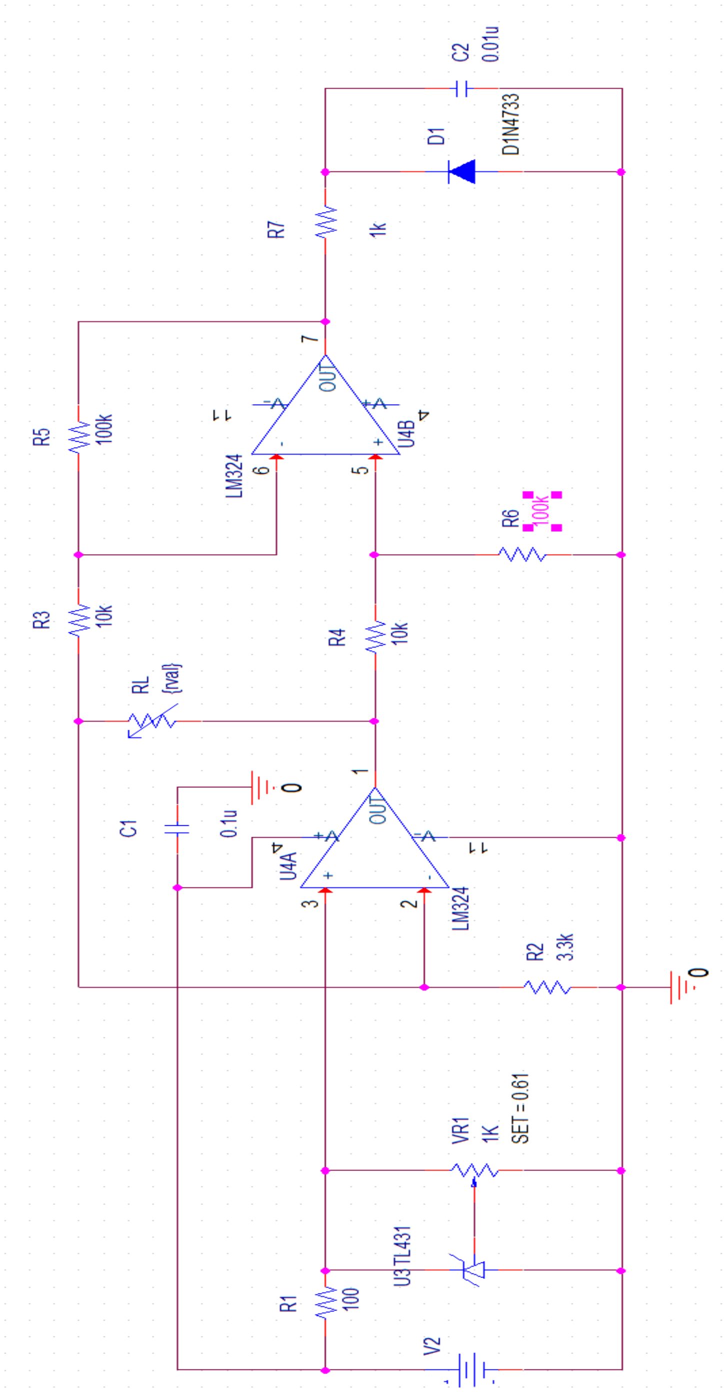


Figure 5 Signal conditioning circuit

In Figure 5, it shows signal conditioning circuit design.

From the left (lower) side of RL, the function of the module is to generate the constant current through the RL (PT100).

The TL431 is a voltage regulator, it controls the constant voltage to input into amplifier U4's pin 3. The input voltage should be 4.096V. This can be reached by adjust the trimmer VR1.

The amplifier gate A takes charge of controlling constant current through PT-100 R_L. The current value is calculated by using the input voltage of U4A divided by R2.

$$I_{RL} = V_{U3} \div R_2 = 4.096 \div 3.3k = 1.2mA$$

However, in practice, the amplifier circuit cannot reach the goal of generate the ideal constant current. If current fluctuation can be controlled in a small range, it can be seen as constant.

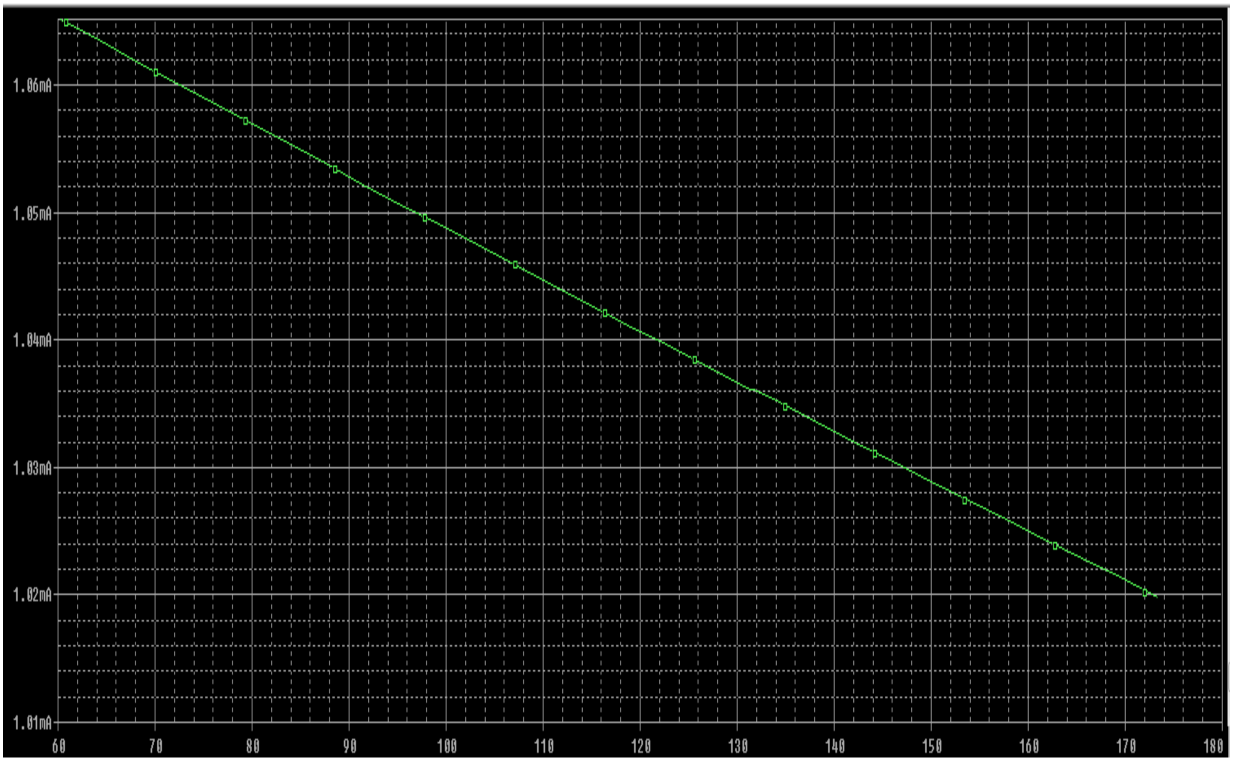


Figure 6 current through the PT-100

In Figure 6, X-axis shows the values of PT-100 in ohms. Y-axis shows the values of currents through PT-100 in mA.

The maximum variation of the current is around 0.045mA. It means the current change is so small that can be ignored. So it is verified that the current through the R_L (PT-100) is constant.

From the right (upper) side of R_L, the module function is amplifying the voltage over RL.

The amplifier gate B U4B amplifies the voltage over the PT-100 RL. The amplified gain is calculated using R5/R3 (R5=R6, R3=R4 is differential amplifier circuit required).

$$A = R_5 \div R_3 = 100k \div 10k = 10$$

In Figure 7, it shows the final output voltage variation refer to the RL (PT-100).

Figure 7 Output voltage of temperature sensor

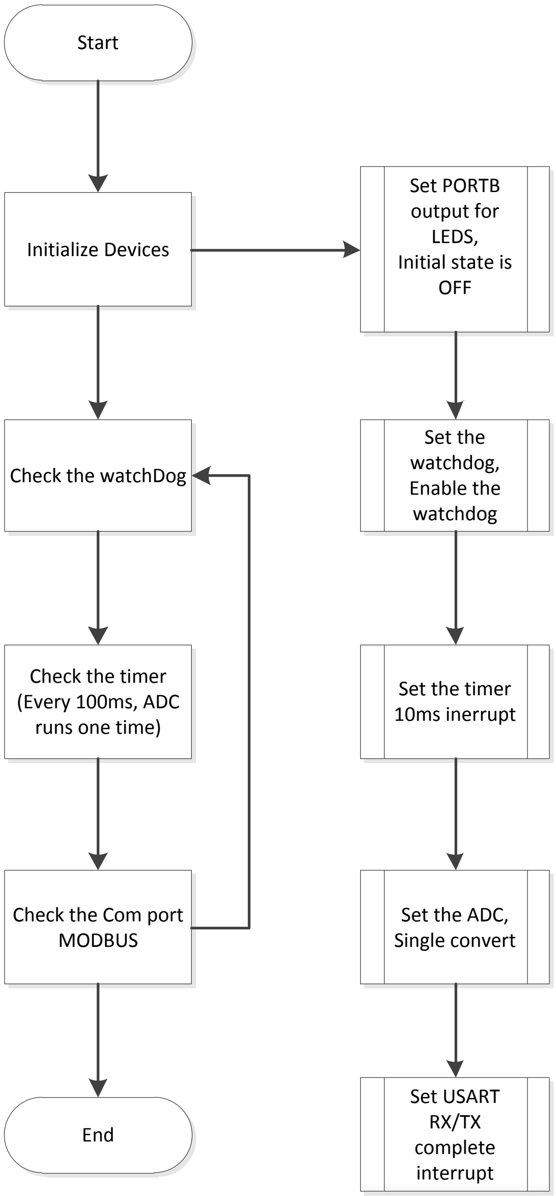
In Figure 7, X-axis also shows the values of PT-100 in ohms. Y-axis shows the values of the output voltages.

The voltage variation is 0.8V during the interval from 70 ohms to 150ohms. The resistance interval corresponding to the temperature is between -60 and 120℃. It is known that the temperature measured is proportional to the resistance of sensor PT-100.

The PCB design is done by the PADS LOGIC and PADS. The final CAM file is in appendices.

5.2 Digital signal processing in STK501

In the STK501, digital signal processing is usually to convert the signal from an analog to a digital form, by sampling and then digitizing it using an analog-to-digital converter (ADC). This digital signal then will be packaged in frame to send it to the laptop using USART. The frame format is according to its function and the protocol MODUBS.



FlowChart 1 MCU operation process

FlowChart 1 shows the microcontroller operation process. The process mainly includes the analog to digital convention and data transition.

5.2.1 Timer

Timer is typically digital counters that either increment or decrement at a fixed frequency, which is often configurable, and that interrupt the processor when reaching zero, or a counter with a sufficiently large word size that it will not reach its counter limit before the end of life of the system.

TCCR0 is Timer/Counter Control Register

TCCR0 = 0x00;

//bit6:3=0 timer select normal mode

//bit2:0=0 no clock source (timer stop)

TCCR0 = 0x03;

//CLK/64 (set prescaler)

//Timer starts

TIFR is Timer/Counter Register

TIFR = 0x00;

//When TIMSK is set, the bit0 set to 1 automatically

TIMSK Timer/Counter Interrupt Mask Register

TCNT0 = 0x83;

// (255-123+1)*64/8000000=1ms

In the initialization, the timer mode, timer prescaler and timer value are set.

Timer interrupt can be started when the TIMSK and global interrupt is set.

Timer interrupt is divided into 3 intervals. They are respectively 1ms, 100ms and 1s.

1millisecond: USART receiving timer-out is checked. If there is no data received currently, the receive buffer will be reloaded.

100millisecond: ADC starts once and ADC register value is saved to the ADC buffer when the status of ADC is set ON. Otherwise, ADC will not start.

1 second: The led status is changed with the interval of 1s. This is a test for timer. The timer configuration is right if the led blinks in an interval of 1s which can be examined by a stop watch. In the meantime, the blinking led also shows that the Microcontroller starts working!

5.2.2 USART

USART (universal synchronous/asynchronous receiver/transmitter) is a piece of computer hardware that translates data between parallel and serial forms. It is commonly used in conjunction with communication standards RS232 or RS485.

UCSR0 is USART0 Control and Status Register

UCSR0A = 0x00;

// disable all kinds USART interrupt

UCSR0B = 0x00;

//disable USART while setting baud rate

UCSR0B = 0xD8;

//Bit 7 – RXCIEn: "1" RX Complete Interrupt Enable

//Bit 6 – TXCIEn: "1" TX Complete Interrupt Enable

//Bit 4 – RXENn: "1" Receiver Enable

//Bit 3 – TXENn: "1" Transmitter Enable

UCSR0C = 0x06;

//Bit 6 – UMSELn: "0 "USART Mode Select "Asynchronous Operation"

//Bit 5:4 – UPMn1:0: "0" Parity Mode "None"

//Bit 3 – USBSn: "0" Stop Bit Select "1 bit"

//Bit 2:1 – UCSZn1:0: "11" Character Size 8 bit

UBRR0 is USART Baud Rate

Registers

UBRR0L = 51;

//set baud rate low

UBRR0H = 0x00;

//set baud rate high

// when fosc = 8MHz, UBRR0 setting value is 51

In the initialization, the transmitter and receiver, transmitter and receiver complete interrupt are enabled.

The receiving index led is turned on.

System required serial port settings in USART as below:

Baud rate: 9600

Char size: 8 bit

Stop bit: 1 bit

Parity: None

When USART receiving completes, the interrupt is triggered. Then the data in USART register will be saved to receiving buffer in sequence. The maximum size of buffer is 32byte. The receive buffer will be reloaded, if the USART is timeout in 10ms.

The MCU will configure the sending buffer after the USART receiving completes. Then the data in the sending buffer will be transferred to the USART register in sequence. This will make the USART send the data.

After all the data in the sending buffer are sent, the receive status will be restored.

5.2.3 Watchdog

Watchdog is a separate counter counts if it starts. It will force the system stop, if the program is stuck.

```
wdt_reset();

 //this prevents a timout on enabling

 wdt_enable(7);

 //WATCHDOG ENABLED – don’t forget to issue WDRs
```

Watchdog prescaler chosen is 2048 KHz. The timeout counts almost 2s. If watchdog is not fed in 2s, the program is forced to stop.

5.2.4 ADC

Analog to Digital Converter (ADC) transform the temperature analog signal to digital format in every 100ms if the ADC status is ON.

ADMUX is ADC Multiplexer Selection Register

ADMUX=0xDE;

//ADMUX = 0x00;

//Bit 7:6 – REFS1:0: "00" Voltage Reference Selection "AREF, Internal Vref turned off"

//Bits 4:0 – MUX4:0:"00000" Analog Channel and Gain Selection Bits"ADC0-PF0"

ADCSR is ADC Control and Status Register

ADCSRA = 0x85;

//Bit 7 – ADEN: "1" ADC Enable

//Bit 5 – ADFR: "0" ADC Free Running Select" no free run, single mode"

//Bits 2:0 – ADPS2:0: "101" ADC Prescaler Select Bits"32"

ADCSRA = 0x00; //disable ADC

In the ADC initialization, the reference voltage is set to 5V. The single mode is selected. PF0 receive the input temperature analog signal.

rpt100 = 5*(float) tempADC /1024/beta;

adcData[adcCount]=(uint16_t)(1/0.385*rpt100-100/0.385)+100;

//temperature from -100 to 200

rtp100 is the resistance value of the PT-100.

tempADC is the value of ADC register.

adcData is the ADC buffer.

The beta value should be the amplifier gain times the constant current in the signal conditioning circuit. The theoretical value is 0.01 (10*0.001A).However, because constant current and amplifier produce the error in practice, the beta value should be adjusted in the calibration step.

The data in the ADC buffer is prepared to be transmitted through the USART. So only the unsigned data type is accepted. Since the temperature value ranges from -60 to 120, the data in ADC buffer or sent data is arranged to add 100 in case of producing negative data.

The temperature data saved in the ADC buffer is calculated according to the formula below:

$$R_{\mathrm{T}} = 0.385 * \Delta \mathrm{T} + R_{T0}$$

This formula shows the relationship between temperature and resistance of PT-100, which has already introduced in the previous analysis.

5.2.5 PORT

PORTB = 0xFF;

//set default 7 LEDs off

DDRB = 0xFD;

//Pin direction is output for “1”

PORTB is reserved for the index led. It is connected to the LEDS port.

Led 7 off is send enable.

Led 7 on is receive enable.

Led 6 blinked is test for the timer.

Led 5 blink is ADC enable.

5.2.6 Modbus-RTU protocol settings

When the USART check receiver buffer, it will determine what the slave address, function name, register or coil start address, read/write data bytes and CRC code are. The received data frame is transmitted from the laptop.

In general, the function 01, 03, 05, 06, 15, 16 are always used. The meaning of the function can be revealed from its name. The function name is shown as below:

Function 1: Read Coil Status

Function 3: Read Holding Registers

Function 5: Force Single Coil

Function 6: Preset Single Register

Function 15: Force Multiple Coils

Function 16: Preset Multiple Registers

The Function 3 and Function 5 are mainly used in our system. The Function 3 takes charge of requesting for the temperature data. The Function 5 changes the status (ON/OFF) of ADC.

After all frames in the receiving buffer are checked, the response frame will be set and transmitted.

Function 03:

Request frame:

Slave Address	Function code	Start address High	Start address Low	No. of register High	No. of register Low	CRC High	CRC Low
01	03	00	00	00	01	0A	84

This command is requesting the content of ADC holding registers #0 from the sensor slave device with address 01.

01: The slave address

03: The Function Code (read ADC Holding Registers)

0000: The Data Address of the first register requested

0001: The total number of registers requested.

0A84: The CRC (cyclic redundancy check) for error checking.

Response:

Slave Address	Function code	No. of data bytes	data register High	data register Low	CRC High	CRC Low
01	03	02	00	15	8B	79

This response returns the ADC value in register #0 with device #01

01: The Slave Address

03: The Function Code (read ADC Holding Registers)

02: The number of data bytes to follow (1 registers x 2 bytes each = 2 bytes)

0015: The contents of register 0(0x0015 = 21)

8B79: The CRC (cyclic redundancy check).

Function 05:

Request:

Slave Address	Function code	Start address High	Start address Low	Status to write High	status to write Low	CRC High	CRC Low
01	05	00	00	00	00	CA	CD

This command is writing the contents of discrete coil #0 to ON and the slave address is #01.

01: The Slave Address

05: The Function Code (Force Single Coil)

0000: The Data Address of the coil.

FF00: The status to write (FF00 = ON, 0000 =OFF)

CACD: The CRC (cyclic redundancy check) for error checking.

Response:

Slave Address	Function code	Start address High	Start address Low	Status to write High	status to write Low	CRC High	CRC Low
01	05	00	00	00	00	CA	CD

The normal response is an echo of the query, returned after the coil has been written.

01: The Slave Address

05: The Function Code (Force Single Coil)

0000: The Data Address of the coil.

FF00: The status to write (FF00 = ON, 0000 =OFF)

CACD: The CRC (cyclic redundancy check) for error checking.

5.3 GUI application on master station

This application is designed using the Qt language which is specialized in GUI designing. Graphic user interface is convenient to monitor the remote terminal interface (RTU). In Table 2, the following functions are realized in GUI application.

Function	Descriptions
Serial Port	Configure the serial port
MODBUS	Configure the Modbus frames
MODBUS Monitor	Display the send/receive raw frame
Register	Display the received data in register

DataBase	Configure the mySQL database

Table 2 Descriptions of functions in GUI application

5.3.1 Serial port configuration

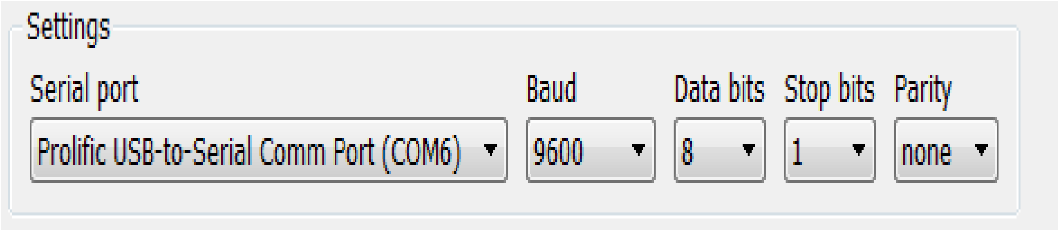


Figure 8 Serial port

In Figure 8, it shows serial port configuration.

When the GUI application is started, it will check the COM port of local computer.

The laptop in this system needs a peripheral of USB-to-Serial cable which will produce a COM port for application.

Baud rate is the value to measure how many symbol per second. Each symbol can represent or convey one or several bits of data. But in serial port, there are only two logic levels (0 and 1). So it means the baud rate value equals to the bit rate value which indicates the transmission speed. The system set the baud rate to 9600 baud which is required for the MODBUS protocol.

Data bits' value in each character can be 5 (for Baudot code), 6 (rarely used), 7 (for true ASCII), 8 (for any kind of data, as this matches the size of a byte), or 9 (rarely used). 8 data bits are almost universally used in newer applications.

Stop bits sent at the end of every character allow the receiving signal hardware to detect the end of a character and to resynchronize with the character stream. In the electronic device, 1 stop bit is always set.

Parity is a method of detecting errors in transmission. The electronic device adds up the number of bits in a character, and if the parity bit setting disagrees with the sum of the other bits, the device reports an error. Parity-checking schemes work by storing a one-bit digit (0 or 1) that indicates whether the sum of the bits in a data item is odd or even. In this system, none is set which means no parity bit. This setting is approved in MODBUS protocol.

5.3.2 Modbus request settings

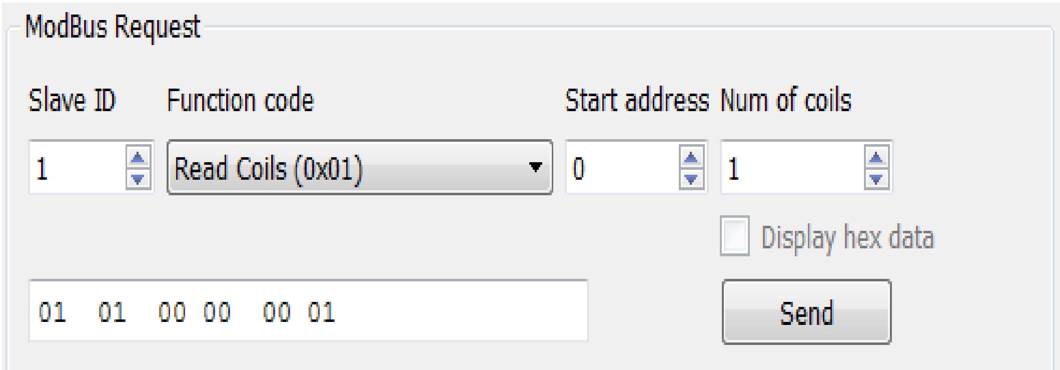


Figure 9 Modbus request setting

The Figure 9 shows the MODBUS settings which will contribute to research about MODBUS protocol.

Slave is the request address. It always take the first byte in the transmit/receive frame.

Start address is the bytes to let the slave know from where of their register to send the data.

Number of coils is how long bytes of the data to be sent. From the start address and the number of coils, the required data can be found in the register.

Function code is the request command. It tells what the slave should do. Several commonly used functions are introduced as below.

1. Function 03 - Read Holding Registers

Request

Function code 1 Byte: 0x03

Starting Address 2 Bytes: 0x0000 to 0xFFFF

Quantity of Registers 2 Bytes: 1 to 125 (0x7D)

Response

Function code 1 Byte: 0x03

Byte count 1 Byte: 2 x N*

Register value: N* x 2 Bytes

(*N = Quantity of Registers)

Example used in System:

Send data: 01 03 0000 0002 c40b

Received data: 01 03 04 00 04 00 07 fa30

2. Function 05 - Write Single Coil

Request

Function code 1 Byte: 0x05

Output Address 2 Bytes: 0x0000 to 0xFFFF

Output Value 2 Bytes: 0x0000 or 0xFF00

Response

Function code 1 Byte: 0x05

Output Address 2 Bytes: 0x0000 to 0xFFFF

Output Value 2 Bytes: 0x0000 or 0xFF00

Example used in System:

Send data: 01 05 0000 ff00 8c3a

Received data: 01 05 0000 ff00 8c3a

3. Function 01 - Read Coils

Request

Function code 1 Byte: 0x01

Starting Address 2 Bytes: 0x0000 to 0xFFFF

Quantity of coils 2 Bytes: 1 to 2000 (0x7D0)

Response

Function code 1 Byte: 0x01

Byte count 1 Byte: N*

Coil Status n Byte: n = N or N+1

(*N = Quantity of Outputs / 8, if the remainder is different of 0 ⇒ N = N+1)

4. Function 06 - Write Single Register

Request

Function code 1 Byte: 0x06

Register Address 2 Bytes: 0x0000 to 0xFFFF

Register Value 2 Bytes: 0x0000 to 0xFFFF

Response

Function code 1 Byte: 0x06

Register Address 2 Bytes: 0x0000 to 0xFFFF

Register Value 2 Bytes: 0x0000 to 0xFFFF

5.3.3 MCU control box's function

There are two buttons specialized to control the RTU. One button reads the temperature data from certain sensor salve. The other writes to change the status of ADC (ON or OFF). If the ADC is in the state of ON, there is temperature data to be read. If the ADC is in the state of OFF, the request will get null feedback.

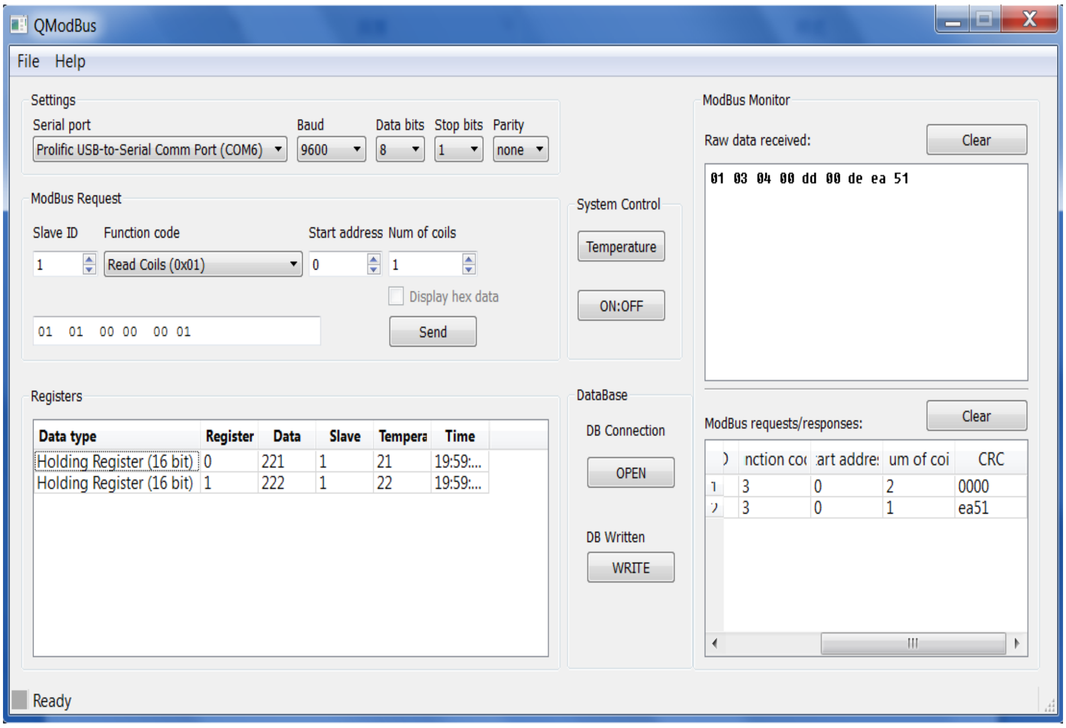


Figure 10 Get temperature value

In Figure 10, if the Temperature button in System control group box is pressed, the received and send data will be displayed respectively in the table of Register, text edit of Raw data received and table of ModBus request/ responses.

In the table of Register, the useful information of received data will be displayed in category. The first column is type of received data. 16 bit means the maximum displayed value is 65535 (2^16-1). The second column is register address which means where the received data are read. Third column is received temperature data which is not formatted. The fourth column is the slave address. The fifth column is the formatted temperature value that can upload to Database. The final column is the time to read the sensor. The rows show all the useful information of received data of different register address.

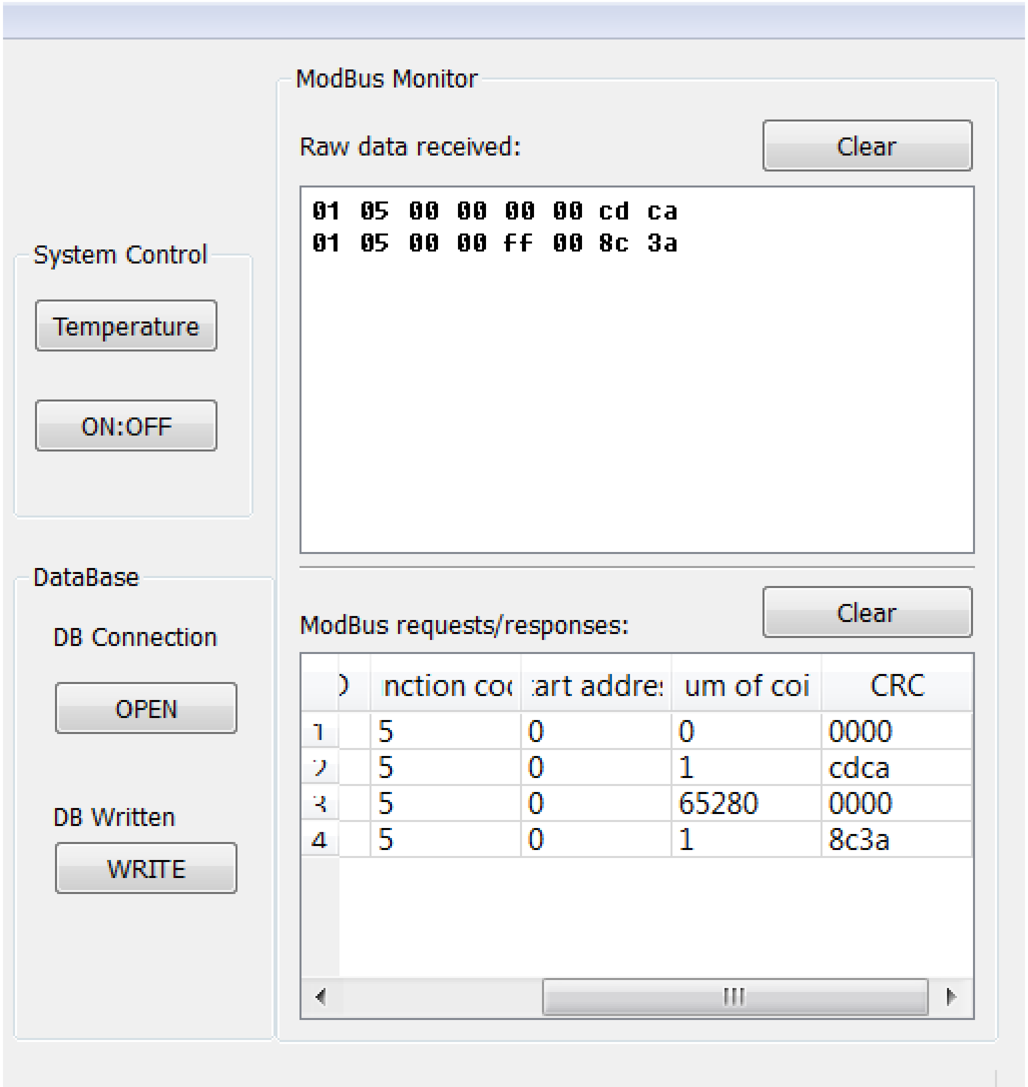


Figure 11 MODBUS monitor

Figure 11 shows that the ModBus monitor group box can display the sent and received data frame.

In the text edit of Raw data received, it displays one received data frame for each row.

In the table of ModBus request/ responses, it shows the structure information of the received and transmitted data frame in category.

In Figure 11, the ON: OFF button has been pressed twice in the example. The ModBus request/ responses shows that the function code is 5, the register start address is 0, the number of coils is 1 and the CRC can be checked. The Raw data received shows the status of ADC. If counting from left to right for the raw data received, the bit 8 to bit 11 are 0000 and FF00 respectively. 0000 means the ADC is in the state of OFF, while FFFF means the ADC is in the state of ON.

5.3.4 MySQL database control box's function

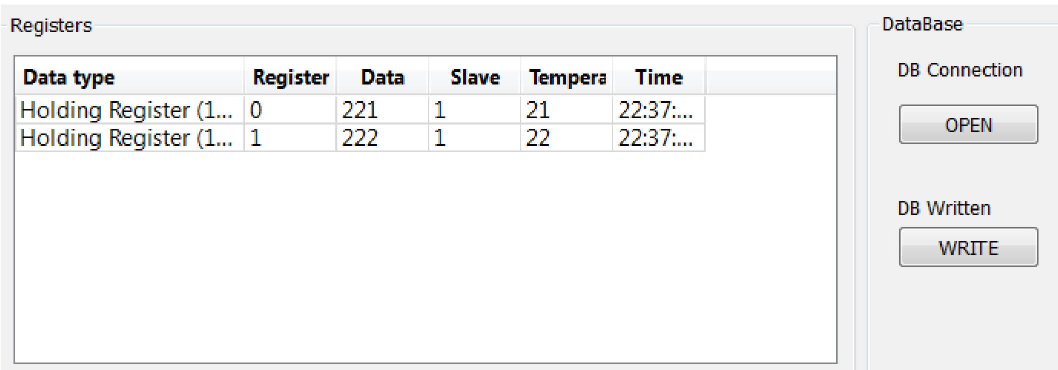


Figure 12 Database control

In Figure 12, database group box has two buttons. The button OPEN starts the mysql database. It sets up the connection between the mysql database and the laptop and initializes the table which saves the temperature data information in Database. The button WRITE writes data to the database from the row 1 of table Register. The data includes information of slave address, temperature and time. Except for these, the states of the ADC are still to be written. The below is mysql query language:

```
query1.exec ("drop table temp");
```

//throw away the table temp if it exists

query1.exec("create table temp(ID int(100) primary key,Slave varchar(15),SwitchStatus varchar(15),Temperature varchar(15),Time varchar(30))");

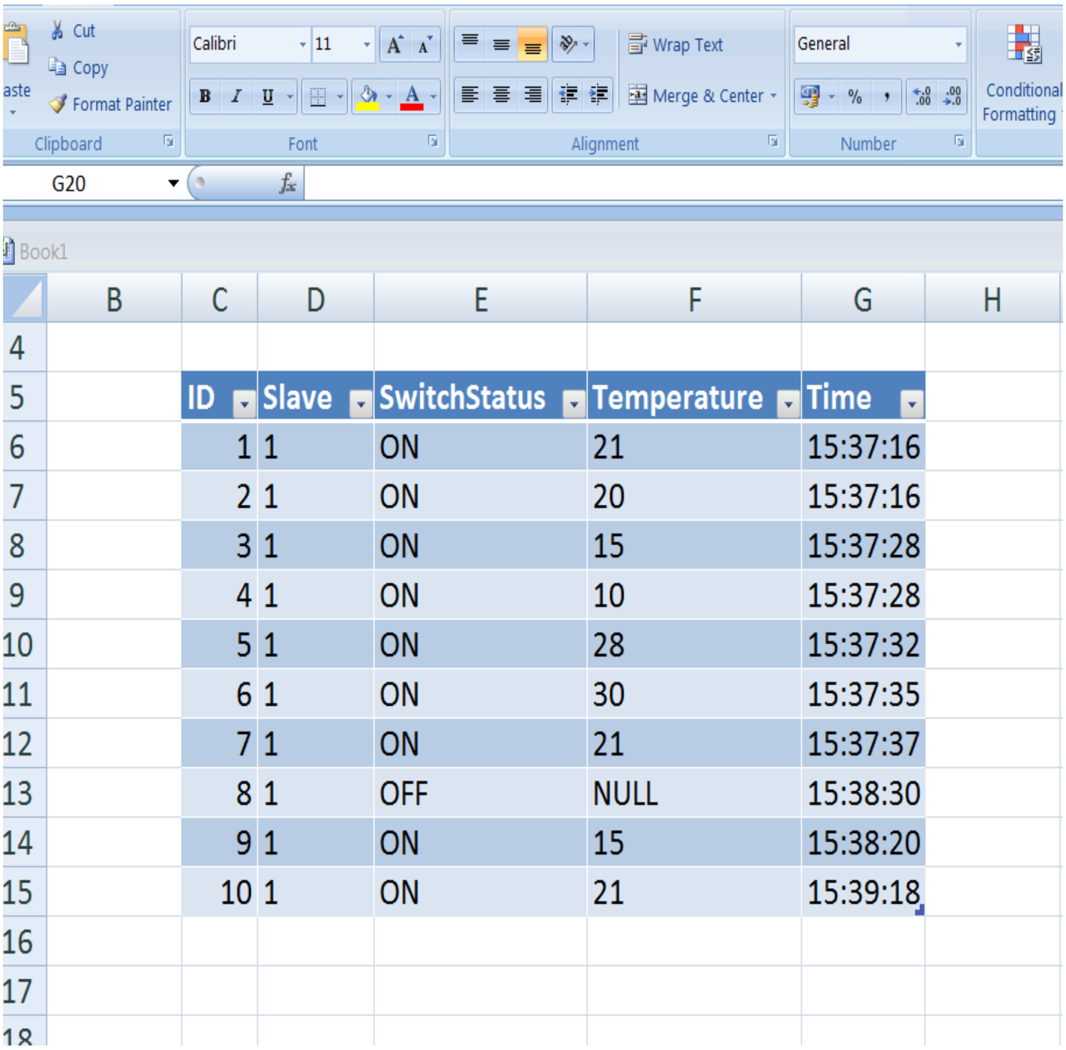
// create table temp

QString tableData=QString("insert into temp values(%1,'%2','ON','%3','%4')").arg(dataCount).arg(slave_reg).arg(temprature_reg).arg(time_reg);

query2.exec (tableData);

//add the values in the table

5.4 Excel displays data from the MySQL database



ID	Slave	SwitchStatus	Temperature	Time
1	1	ON	21	15:37:16
2	1	ON	20	15:37:16
3	1	ON	15	15:37:28
4	1	ON	10	15:37:28
5	1	ON	28	15:37:32
6	1	ON	30	15:37:35
7	1	ON	21	15:37:37
8	1	OFF	NULL	15:38:30
9	1	ON	15	15:38:20
10	1	ON	21	15:39:18

Figure 13 MySQL data in Excel

Figure 13 shows the Database table named temperature in the Excel.

In the table, slave address is displayed in the second column; the state of ADC is displayed in the third column; the temperature is displayed in the fourth column and the time is displayed in the fifth column.

Before Excel are connected to mysql database, the ODBC need to be installed. The mysql ODBC configuration is shown in Figure 14 and Figure 15.

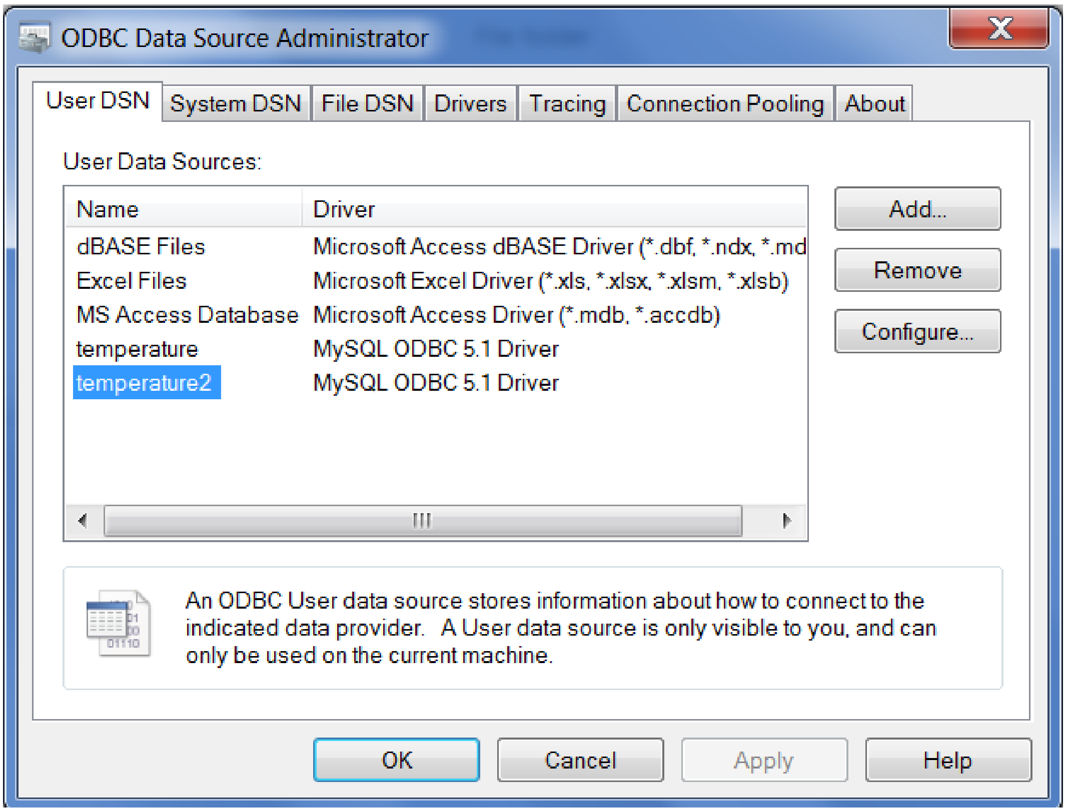


Figure 14 ODBC data source administrator

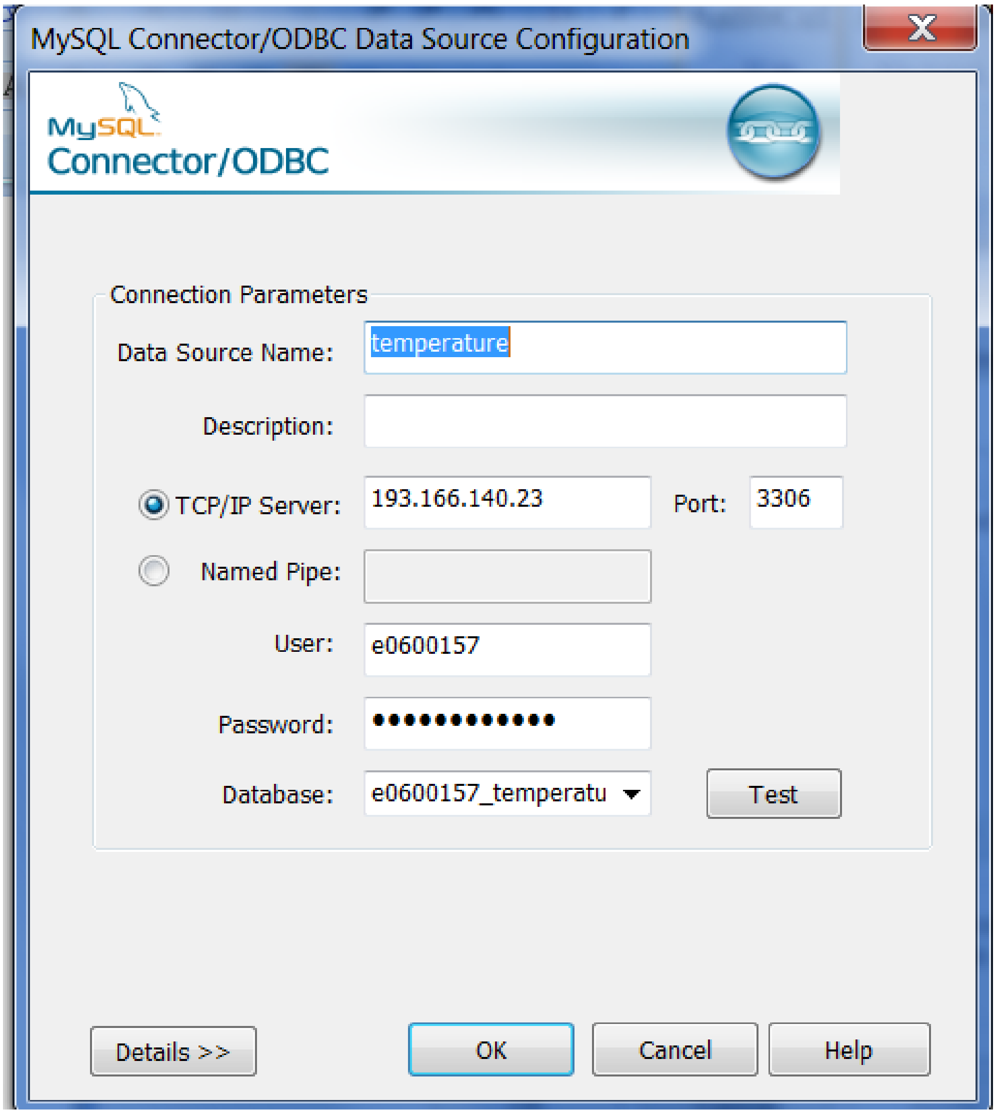


Figure 15 ODBC connector

It should be noticed that Excel 2007 just supports the 32bit mysql ODBC. So only the 32bit ODBC data source administrator is compatible to configure the 32bit mysql ODBC. In the operating system of windows 64bit, the 32bit ODBC data source administrator application is under the folder of C:\Windows\System32. Other ODBC data source administrator cannot find the mysql ODBC driver.

6 TEST

The system should be set up before the test as follows.

1. Connect PCB to the 5V voltage source.
2. Connect the 5V voltage source to STK501 AVREF.
3. Connect PCB output pin to STK501 ADC input pin.
4. Turn on the STK501 power source.
5. Connect the STK501 to computer using RS232.

The system can be controlled in the GUI application as Figure 10, after it is set up well.

The COM port will be detected automatically when the GUI application is started.

In Figure 10, the serial port settings are as shown. Otherwise, an exception error window is popped up.

Then the way of testing follows the steps shown below:

1. In “Modbus request” block, the function 0x03 and function 0x05 will be selected to test responses from the sensor slaves. The received data can be read in the “Register” block and raw data received and its frame structure data is shown in “Modbus Monitor” block.(Other functions are reserved for future use.)
2. The button “Temperature” controls to read the temperature data from sensor. And they are shown in row 1 and column 5 of “Register” block.
3. The button “ON: OFF” controls the status of ADC (ON and OFF) and the ADC index led in the MCU (STK501).
4. If the button “OPEN” is pressed, the application will connect to database and refresh the database at the same time (The records in database will be cleared).
5. If the button “WRITE” is pressed, the formatted temperature data and request time shown in “Register” will be saved to the mySQL database at school.

6. Open the ready-made Excel file which is created to show the table temperature in mySQL. The temperature data information can be checked here.

7 DISCUSSION

In this project, some problems are discovered in the process. But they are not solved for some reasons. Some designs are easily misunderstood by users. All of these wonders are discussed as below:

1. The remote terminal unit is not integrated. The sensor conditioning and digital processing devices are separated in two PCB. They are connected through wires, which will take up much more space and waste time in connection. In the meantime, the use of STK501 as MCU has a lot of unused components. All of these will increase the price of system.
2. The GUI application does not apply the function of automatically recording the temperature from the sensor. Some users think that the button “Temperature” and the button “Write” should be merged together and repeat this merged function in a timer interval.
3. The data saved in the database can be shown in the GUI application. It is inconvenient to check in Excel.

The first wonder is a problem caused by the PCB making capability. Incorporating the signal conditioning circuit with MCU is available in the step of PCB design. However, it is not easy for making the PCB by individual. The reason is the lack of professional tools and knowledge.

The second wonder is a misunderstanding. The objective of this project is not only to build up a control system, but also do a research about MODBUS protocol. In this case, every transmitted frame should be monitored in the application. So the functions seem to be complicated.

The third wonder is also a misunderstanding caused by the objective of project. The magnitude of the data record should be considered in the design. If the data in database is large enough, it is inconvenient to show them in the GUI application. In addition, the objective of project does not include displaying the data in database. The Excel work is just an adjunct to check the database value.

8 CONCLUSION

8.1 Project development

This project also can be expanded to the other fields like humidity or pressure monitoring. Other control functions can be added according to the specific work.

For example, in coolant temperature system, if the temperature on a device is too high or too low, a PWM wave can be generated to control the speed of fans so as to control the temperature on the device.

This system developed is only a prototype to learn more about the control system.

8.2 Summary

In this project, the goal is to achieve concrete evidence that this sensor measurement method and communication method of electronic devices can be used and implemented to the industrial applications as one developed here. We just design, develop and construct a prototype system based on the control system technology.

By accomplishing the project work, I am gradually familiar with PCB design, embedded system programming, Qt language and use of MySQL database. There are a lot of difficulties during the development of my work. Since I have no much prior experience with practical control system engineering, the difficulties need to be solved by reading and referring to many related materials in this field, during which skills and knowledge can be increased.

The system can be still improved in any aspects. I would like to see the development of the actual system go ahead in future.

9 REFERENCES

/1/ Positive Analog Feedback Compensates PT100 Transducer. Accessed 22.2.2012. http://www.maxim-ic.com/app-notes/index.mvp/id/3450>

/2/ Atmetga 128L datasheet. Accessed 18.2.2012.

http://www.atmel.com/Images/doc2467.pdf>

/3/ STK501 user manual. Accessed 18.2.2012.

http://www.atmel.com/Images/doc2491.pdf>

/4/ STK502 temperature monitoring system with LCD. Accessed 23.2.2012.

http://www.atmel.com/Images/doc2529.pdf>

/5/ MODBUS tutorial. Accessed 23.2.2012.

http://www.lammertbies.nl/comm/info/modbus.html>

// Qt serial port library configuration. Accessed 23.2.2012.

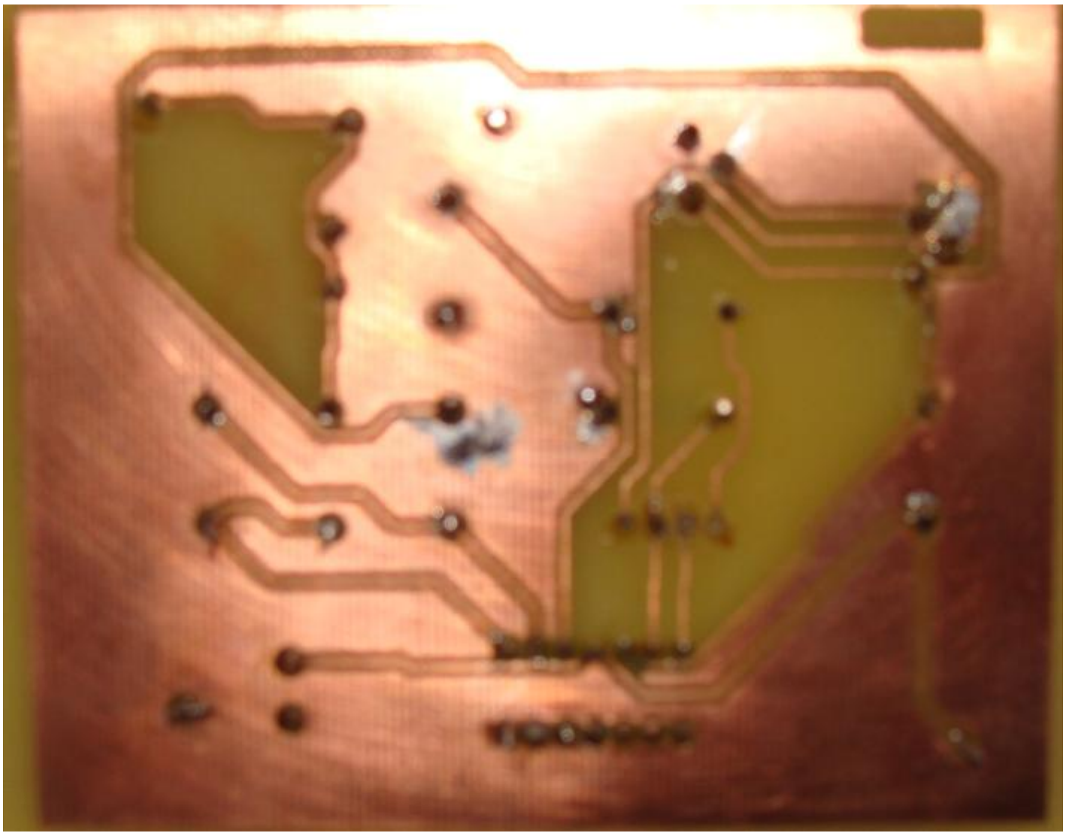
http://code.google.com/p/qextserialport/

/6/ Qt 4.5 and MySQL-plugin with Mingw on Windows XP. Accessed 21.2.2012.

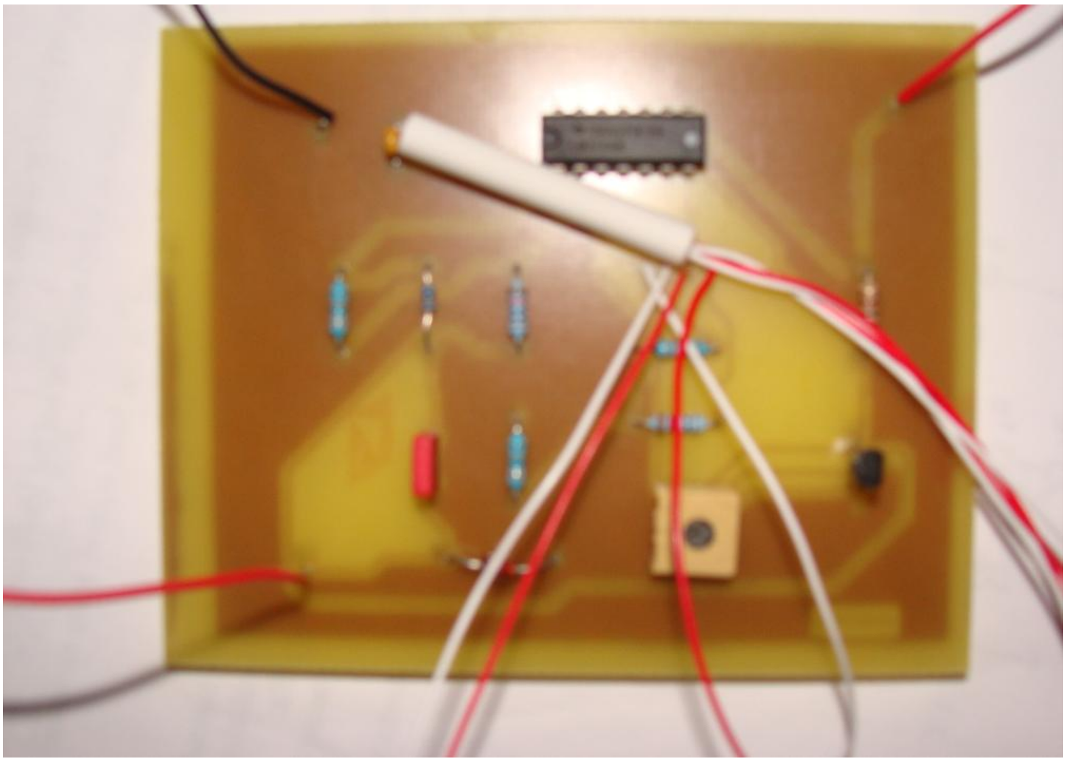
http://christopher.rasch-olsen.no/2009/04/14/qt-45-and-mysql-plugin-with-mingw-on-windows-xp/

APPENDICES

APPENDIX1. Ready-made PCB



APPENDIX2. Components distribution



APPENDIX3. Microcontroller Connection