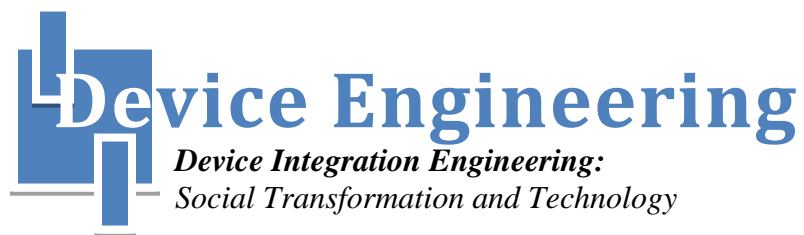


Lhujha2011

Rajendra Man Banepali

<http://rajendrabanepali.wordpress.com>



Contents

1	FOREWORD.....	3
2	MISSION OF PROJECT:.....	3
3	SCOPE	3
4	CONFIGURATION AND MECHANISM:	3
5	DESIGN ENGINEERING:	4
5.1	CIRCUIT DESIGN	4
5.2	INPUT	4
5.2.1	<i>Power Supply Module</i>	<i>4</i>
5.2.2	<i>Sensor Module</i>	<i>5</i>
5.3	PROCESSOR.....	8
5.3.1	<i>Microcontroller Configuration</i>	<i>8</i>
5.3.2	<i>Firmware Design</i>	<i>9</i>
5.4	OUTPUT	9
5.4.1	<i>Motor Interfacing Module (Relay and Motor Interfacing).....</i>	<i>10</i>
5.4.2	<i>Serial Communication (RS232) Module</i>	<i>11</i>
5.5	COMPLETE CIRCUIT.....	12
5.6	COMPLETE PICTURE OF DEVICE	13
6	ENHANCEMENT AND INTEGRATION:	14
6.1	PROTOCOL DESIGN – COMMUNICATION CONTROL PROTOCOL (CCP)	14
6.1.1	<i>Firmware.....</i>	<i>14</i>
6.1.2	<i>Software Application Protocol</i>	<i>14</i>
6.2	WEB ENGINEERING	14
6.2.1	<i>Scope of web application:.....</i>	<i>14</i>
7	RECOMMENDATION	14
8	ANNEX	15
8.1	LHUIHA FIRMWARE IN ASSEMBLY LANGUAGE	15
8.2	LHUIHA FIRMWARE IN 'C' LANGUAGE	17

1 Foreword

Project Warrior V1.0 (**Ljujha2011**) is a concept of automation of two water pump: one is reserved tank water pump and roof top water pump. They are automatically operated when it is really needed regarding the condition of water status of flow on tap, and water on respective tank.

2 Mission of project:

The goal of this project is to reduce the problem of unmanaged water situation at home and integration of web technology with embedded system. Normally, in the context of Nepal the electricity is not regular and it causes the dispute of water supply. The device will automatically work as human interference of switching of water dump according to conditions.

The problem of a house keeper will be reduced using the device and s/he can concentrate the time toward the productive work. The technology is indirectly influencing social transformation.

3 Scope

This is the implementation of the embedded system using 8051/2 micro-controller. The device works to automate two water pumps and communicate with computer with web application for further processing.

This document includes documentation of Firmware Design, Circuit Design, and Printed Circuit Board (PCB) Design.

4 Configuration and Mechanism:

Motor A is connected with reserved water tank and Motor B is connected with roof top tank. To run any of the Motor, there should be water flow on the tap. Motor A is started when water flow on the tap, and tank A is not full. Motor B is started if there is water flow on the tap, tank A is not under reserved and tank B is not full.

Four sensors are installed into the tank in such a way they work for micro-controller. The sensor can be designed with copper wire and plastic.

5 Design Engineering:

Regarding hardware Atmel 8051/2 (AT89C51/2) Micro-Controller is used. The firmware is written into assembly and C programming. Printed Circuit Board (PCB) is designed for circuit design of this device.

5.1 Circuit Design

The device's circuit can be divided into three modules: Power Supply Module, Microcontroller and Sensor Module, Motor Interfacing Module (Relay and Motor Interfacing), and Serial Communication (RS232) Module. The modules wise circuits though logically separate here are integrated within one circuit design and PCB. The module wise circuit will be easy to debug the device in debugging.

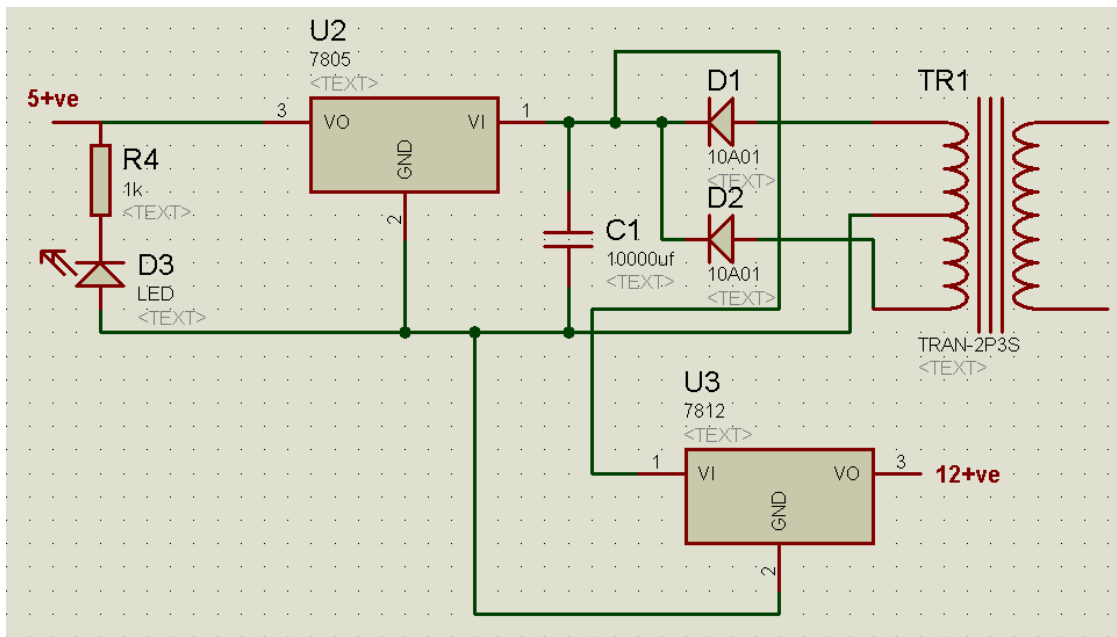
We can divide those modules into three stage of device: Input (Power Supply Module, and Sensor Design), Processor (Firmware, and Microcontroller), and Output (Relay and Motor Interfacing, and Serial Communication as I/O).

5.2 Input

Without input here is interfering of human into device. Input can be interrupt or it may be trigger for any event to do into microcontroller. This device has input section as follow:

5.2.1 Power Supply Module

The module has two voltage regulators: 7805 to produce 5+ve for microcontroller and 7812 to produce 12+ve for relay interfacing and other motor interfacing. The device is designed independently so the complete power supply should be integrated into the device. Necessary transformer and power supply circuit is included into the circuit.



Items used in circuit

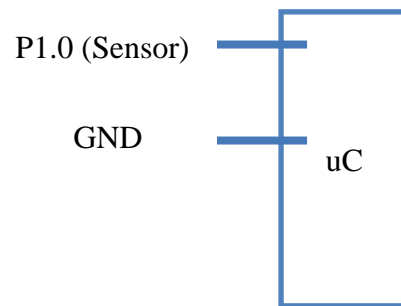
Sn	Item	Description	Qty
1	Adaptor	15-0-15 / Min 500Ma	1
2	Diode	A/C to DC convertor	2
3	Capacitor	10000uf – to control voltage fluctuation and device security	1
4	5volt Regulator 7805	5volt regulator for microcontroller	1
5	Register – 1K	Limiting current to LED	1
6	LED	5volt Indicator	1
7	12volt regulator 7812		1

5.2.2 Sensor Module

Here, the sensor is made by copper rod which is connected with the pin of microcontroller or interfacing device. It is necessary to test some practical experiment for sensor with different mechanism and different way. We have to implement most effective and convenient mechanism into this project. Let's have a look some of test case and experiment of sensor. Those of practical experiments are based on principle of good conduction. We have assumed that water is conduction for current. The sensor can not work on pure water which has not minerals.

5.2.2.1 Test case – using microcontroller pin directly:

The practical experiment was conducted with microcontroller device using pin directly. It is very simple experiment since we have used only bread wire as sensor which is directly connected with microcontroller pin (P1.0). The logic of sensor is that the P1.0 is grounded while the water is crossed though P1.0 and GND Pin. The experiment was not successful since there is no any sense from the wire sensor. We have used only 5volt and 5ma current. The wire cannot sense the current through the water while it was into the water. If we put some salt into the water it is possible to sense the current. But, we need to test the sensor into the clean and drinking water. Microcontroller is based on CIMOS technology and it needs high voltage to sense. So, we need to use Transistor-Transistor Logic (TTL) logic which can sense even a small current.

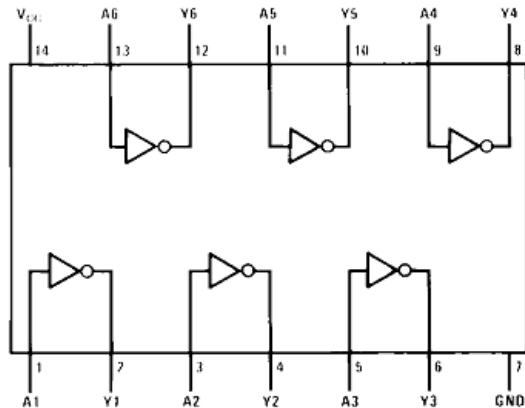


NOTE: The experiment was done base on logical and theory of microcontroller only. The test was failure with given conditions of real world so it is not useful for this project.

5.2.2.2 Test case – using DM7404 Hex Invertor Gates with pull up register:

DM7404 device contains six independent gates each of which performs the logic INVERT (NAN Gates) function. It works on TTL logic since it is more sensitive than Microcontroller.

Connection Diagram



Function Table

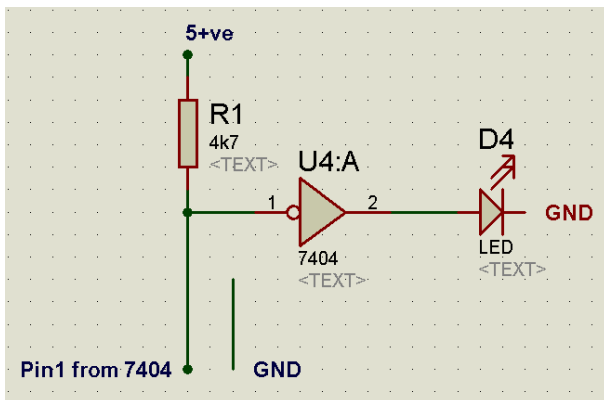
$Y = \bar{A}$

Inputs	Output
A	Y
L	H
H	L

H = HIGH Logic Level
L = LOW Logic Level

DM7404 Hex Inverting Gates

On the lab, we have used DM7404 to test water level. Here, Pin7 is GND and Pin14 is VCC. We have connected a LED to Pin2 (+ve). Pin1 with pull up resistor is used for sensor. When we tested in mineral water, it does not work too. But, it works when we tested with water from normal tank of roof. We tested differently way. Int the mixed water of mineral water and tank water also work but it is bit conductor than original tank water. Pin1 and GND is used to make a sensor. When water drops or flow of water touch those wire, they are open cuircuit and DM7404 send signal to Pin2 as Invert Gate (Please see function table). Here, we got when Pin1 and GND is connection into the water, Pin2 is trigger positively with high voltage as the LED is blink with high voltage. If the LED can blink, the mincrocontroller can be trigger. So, the out put signal Pin2 is connected with mincrocontroller’s input pin. It is only input unit testing case.



The circuit is used in lab for use case. We have used almost 2 meters long wire for sensor. It is nice working when we have put those two wire near by attached. It is necessary to draw serious concertration while designing sensor for practicle experience. So, we have decided to use silver plat so that even tiny current can be sensed by sensor probes.

This experiment works very nicely. We have used 12meter wire to get the sensor to the chips. It works very nice. So, we have decided to use this circuit for sensor.

Listed below has items used into the lab for test case circuit. They are:

Items used in circuit			
Sn	Item	Description	Qty
1	DM7404 Hex Invertor Gates (Do not use 74LS04, use 74HF04)	7407 works on TTL logic and small voltage can work to sense	1
2	LED	Check logic (Blink) if Pin1 is grounded.	
3	Resister 2K/ 4.7k / 5K	Pull up resister to Pin1	
4	Wire – 2m	Wire is connected in Pin1 and GND for sensor	2m

Very Important: use 74HF04 version since I have used LS version but it does not work. When I used HF version, it works perfectly to sense the water.

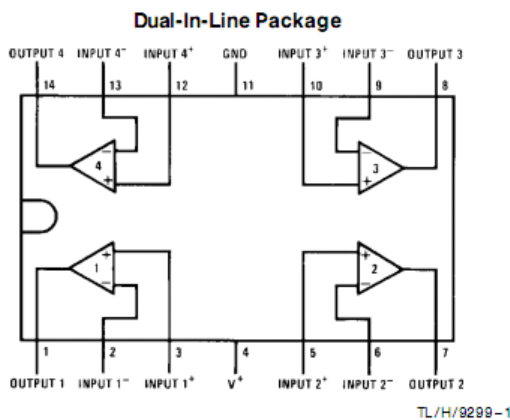
5.2.2.3 Test case – using LM142 /LM342 – Up Amp:

LM124/LM224/LM324/LM2902

Low Power Quad Operational Amplifiers

The LM124 series consists of four independent, high gains internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of

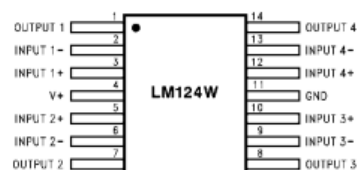
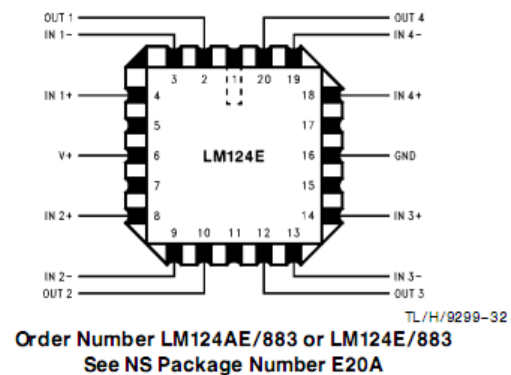
Connection Diagram



Top View

Order Number LM124J, LM124AJ, LM124J/883**, LM124AJ/883*, LM224J, LM224AJ, LM324J, LM324M, LM324AM, LM2902M, LM324N, LM324AN or LM2902N
See NS Package Number J14A, M14A or N14A

*LM124A available per JM38510/11006
**LM124 available per JM38510/11005



Order Number LM124AW/883 or LM124W/883
See NS Package Number W14B

the power supply voltage. Application areas include transducer amplifiers, DC gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems. For example, the LM124 series can be directly operated off of the standard a 5V power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional +-15V power supplies.

5.3 Processor

A microcontroller (sometimes abbreviated μC , uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of NOR flash or OTP ROM is also often included on chip, as well as a typically small amount of RAM. Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, and toys. By reducing the size and cost compared to a design that uses a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to digitally control even more devices and processes. Mixed signal microcontrollers are common, integrating analog components needed to control non-digital electronic systems.

5.3.1 Microcontroller Configuration

MICROCONTROLLER CONFIGURATION:

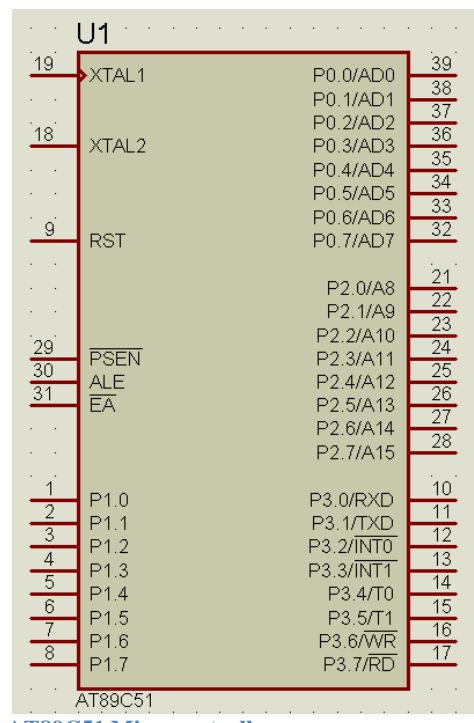
Here, Atmel AT89C51 microcontroller is used for this device. Pins and device details is attached bellow:

Microcontroller: AT89C51

Pins Configuration:

- P1.0 – Motor no. 1: Operation, Status
- P1.1 – Tank no. 1: Full Indicator Sensor
- P1.2 – Tank no. 1: Reserved Sensor
- P1.3 - Motor no. 2: Operation, Status
- P1.4 – Tank no. 2: Full Indicator Sensor
- P1.5 – Water sensor of flow of water on Tap

P1.0 is used for operation of Motor number 1 as switch on and switch off. And the pin can be used for motor status to indicate motor operation using LED. Similarly, P1.3 is used for Motor number 2.



AT89C51 Microcontroller

5.3.2 Firmware Design

Today, in some circles, the word firmware has evolved to sometimes even denote application-level programs stored in NAND flash or NOR flash memory (such as in mobile phones and similar products). However, a more fundamental definition would still be fixed or semi-fixed data in a hardware device. This may include ROM and/or PLA structures for microcode and other data in a processor implementation as well as the low level machine code stored in ROM or flash memory running on that processor (i.e. many products use several levels of firmware). Microcode and other data in an application-specific integrated circuit (ASICs) would also fit that definition very well. The same could be said about programmable logic devices which may have configuration data stored either as internal fuses, in a ROM, or in a flash memory (used much the same way as a ROM or EPROM)

There are no strict boundaries between firmware and software, as both are quite loose descriptive terms. However, the term firmware was originally coined in order to contrast to higher level software which could be changed without replacing a hardware component, and firmware is typically involved with very basic low-level operations without which a device would be completely non-functional. Firmware is also a relative term, as most embedded devices contain firmware at more than one level. Subsystems such as CPUs, flash chips, communication controllers, LCD modules, and so on, have their own (usually fixed) program code and/or microcode, regarded as "part of the hardware" by the higher-level(s) firmware.

Low-level firmware typically resides in a PLA structure or in a ROM (or OTP/PROM), while higher level firmware (often on the border to software) typically employs flash memory to allow for updates, at least in modern devices. (Common reasons for updating firmware include fixing bugs or adding features to the device. Doing so usually involves loading a binary image file provided by the manufacturer into the device, according to a specific procedure; this is sometimes intended to be done by the end user.)

5.3.2.1 *DEVICE Firmware:*

The firmware for this device is called Lhujha Firmware. The firmware is initially designed in assembly language and further enhancement is implemented in C compiler as programme design, testing, and RS232 programming. *Firmware version 1.0 has functionality of four sensor and motor automation only whereas version 1.2 will be enhanced including serial communication (RS232) with web application.* The draft of initial code version 1.0 is in Annex 8.1.

5.3.2.2 *PROTOCOL - DEVICE Firmware:*

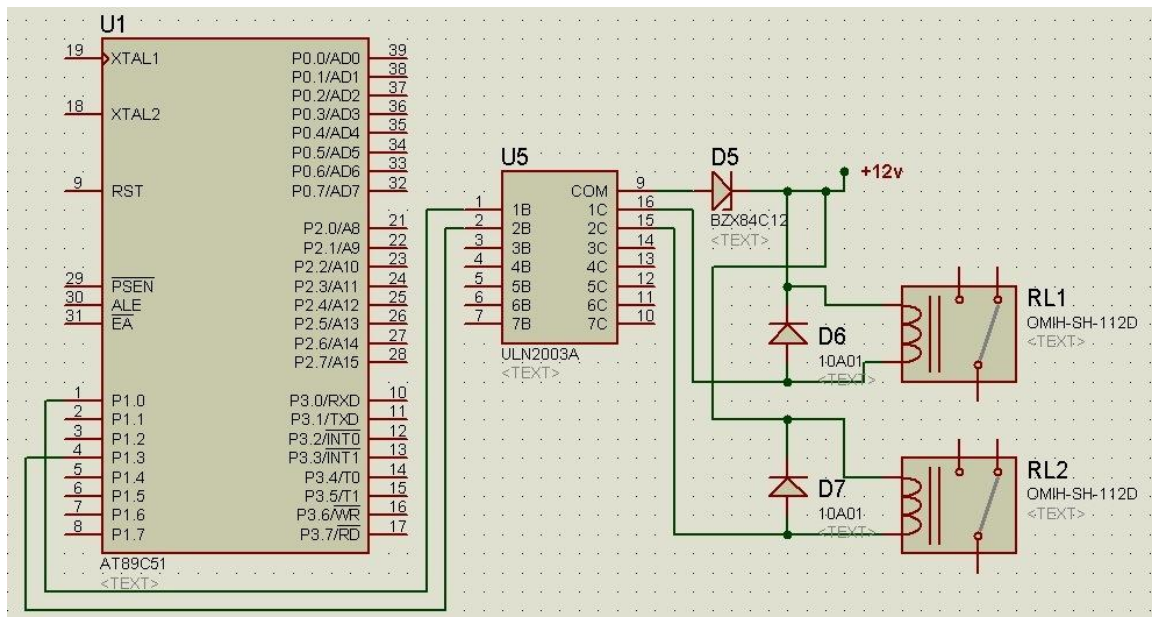
This topic is in detail in 6.1 below.

5.4 Output

Output is view of end of device production. It is based on input and processing business logic. Here, we have very vital output processing: Relay, and Computer interfacing.

5.4.1 Motor Interfacing Module (Relay and Motor Interfacing)

This is output scenario of device for interfacing of motor. There is very simple mechanism to interface the motor. In the context, we have to interface two motor. It means two relay should be trigger by microcontroller. Here, we can use ULN2003 to trigger relay with 12volt. Let's have a look at simple circuit and understand the mechanism.

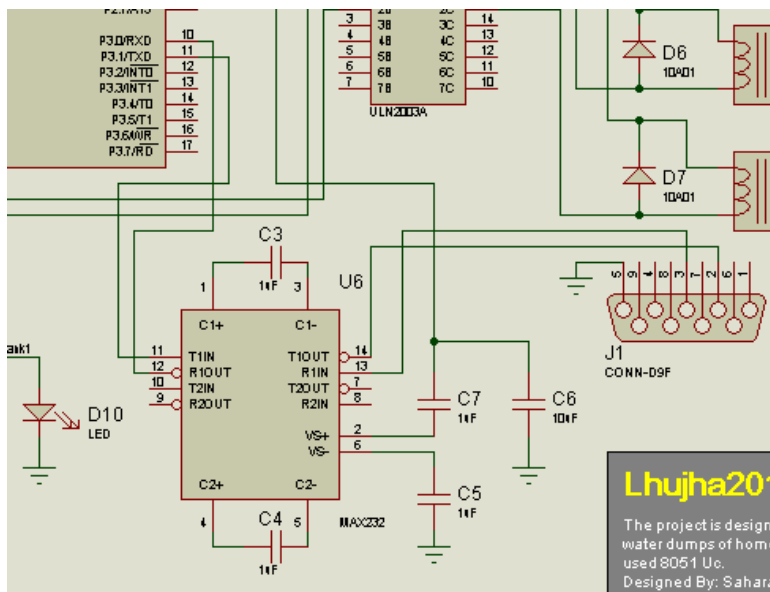


Listed below has items used into the lab for test case circuit. They are:

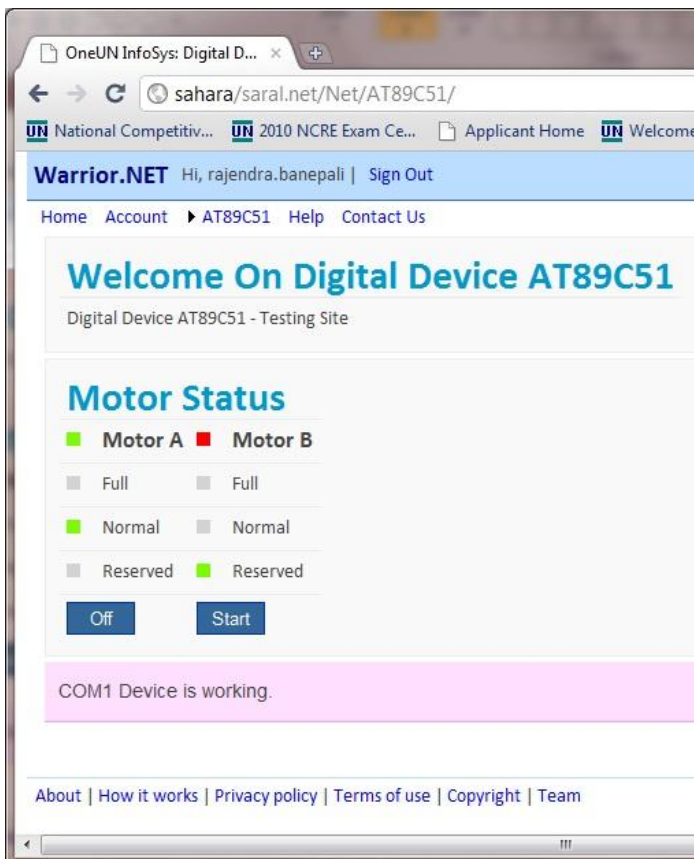
Items used in circuit

Sn	Item	Description	Qty
1	ULN2003	The IC is used to interface switch devices like relay.	1
2	12v Zener Diode	12v Zener diode is used for current protection ULN2003.	1
3	Diode	Relay coils can push back some amount of current back to IC (Generate Back EMF). If high amount of current is drawn from IC ULN2003, the chips can be damage.	2

5.4.2 Serial Communication (RS232) Module

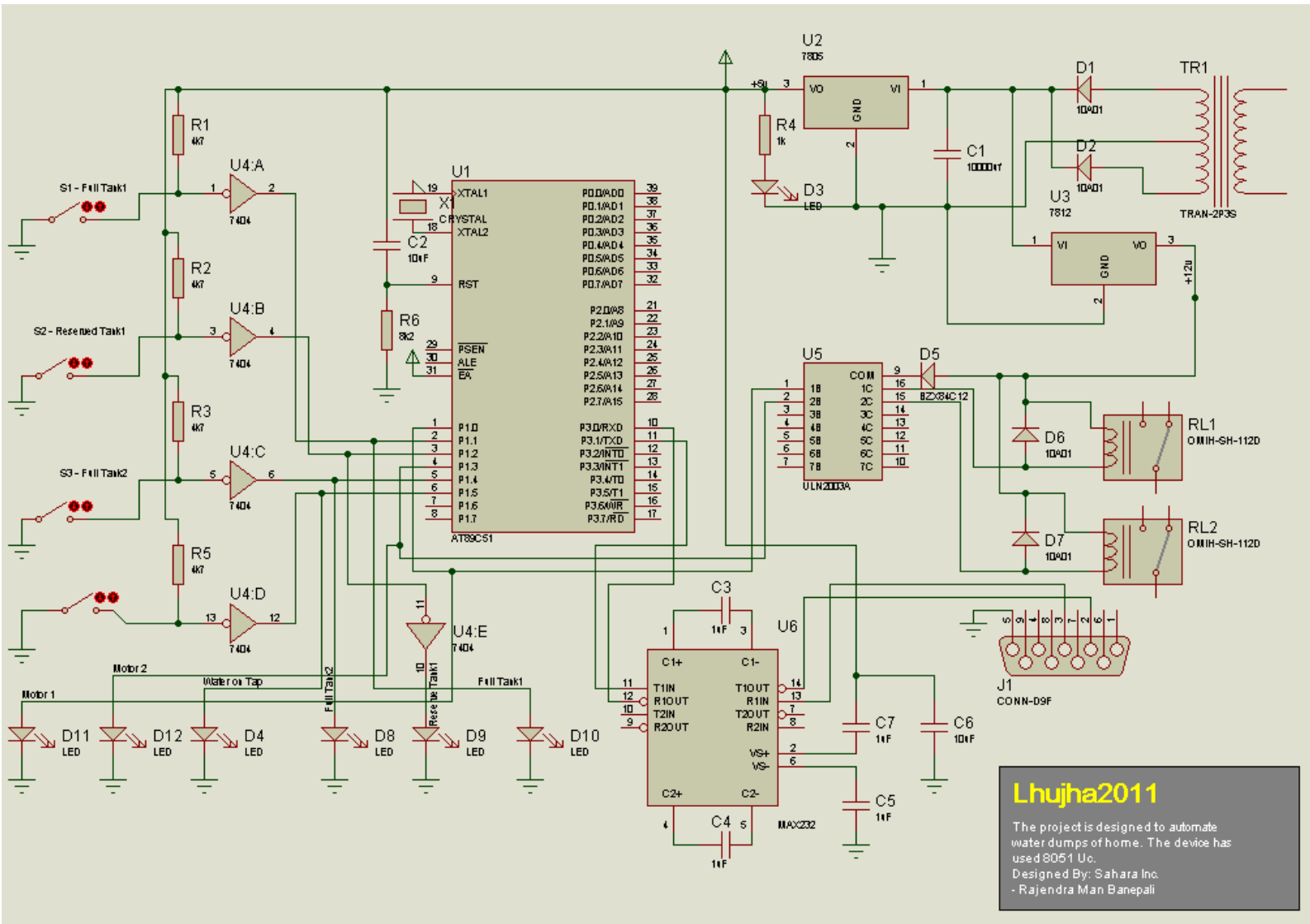


The enhancement of device is to implement serial communication to computer. The device sends operation information to computer via RS232. Web application retrieves the information from the device to present the information on the screen. The screen of computer for information will be as shown below. The web application can operate the motors as they are required.



On the screen, Motor A and Motor B are distinctly visible whereas Motor A is in running stage and Motor B is in rest. In addition, we can view the status of each motor clearly. Buttons are used to operate the motors manually. This is the facility for end user if s/he wants to force to start or stop motors via web application.

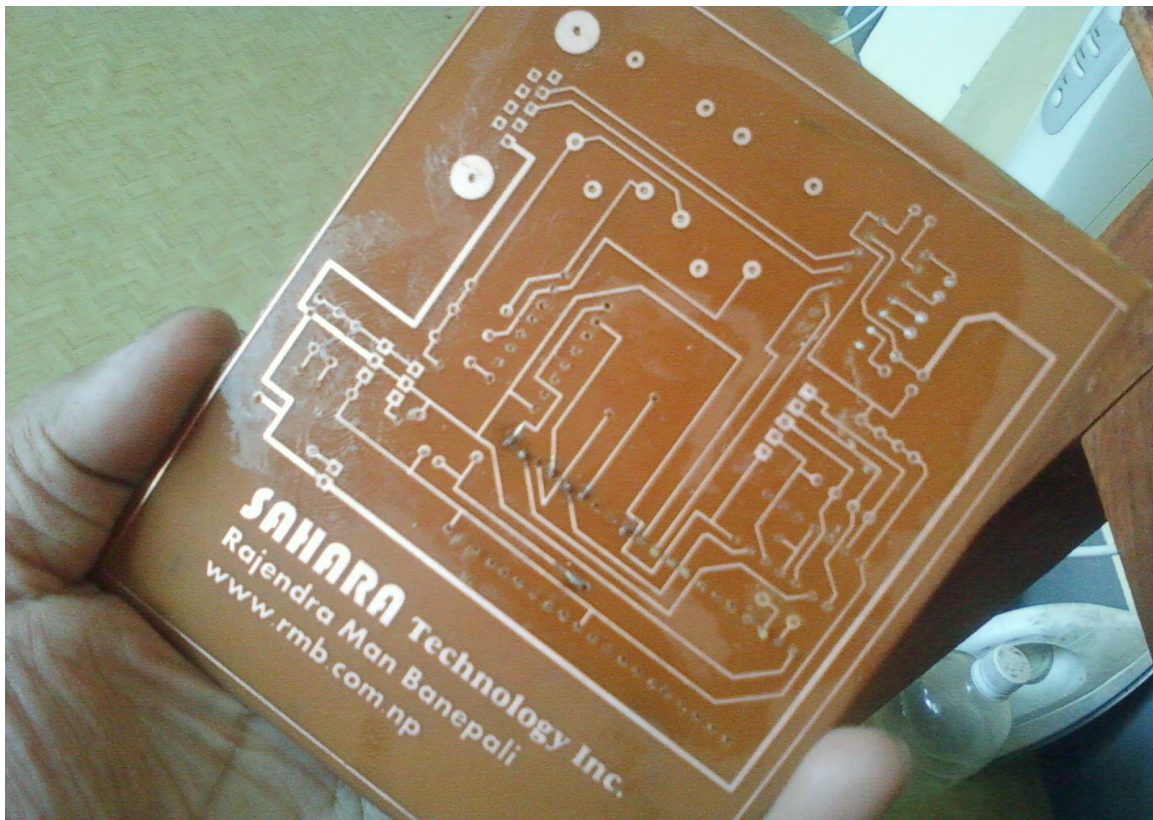
5.5 Complete Circuit



5.6 Complete picture of device



< The device is design by Rajendra Man Banepali adding RS232 communication channel with Computer in standard manner. />



6 Enhancement and Integration:

The device is designed in such a way so that the device can communicate with computer via serial communication RS232. One of the web applications is responsible for communication protocol with the device. ASP.NET or PHP can be used for device communication. I would like to prefer ASP.NET with web service for this project. The computer is connected on internet and the computer acts as web server so that the device status and virtual device representation is appeared into the web so that the device can be controlled via internet/web and mobile.

6.1 Protocol Design – Communication Control Protocol (CCP)

Open to developer for open source – Contact RAJENDRA.BANEPALI@LIVE.COM for expansion.

6.1.1 Firmware

Please see code in 8.1 for firmware in assembly language and 8.2 for firmware in ‘C’.

6.1.2 Software Application Protocol

Open to developer for open source – Contact RAJENDRA.BANEPALI@LIVE.COM for expansion.

6.2 Web Engineering

The device is connected with web application which runs as a service to collect data from the device and operate the device. The web application can get status of Motor: switch state and can operate the Motor as switch on and switch off.

6.2.1 Scope of web application:

Web application is design to view the Motor status and operation of Motor in version 1.x. In further enhancement version 2.x will be integrated with database to record the time and date of device overall operation status.

Open to developer for open source – Contact RAJENDRA.BANEPALI@LIVE.COM for expansion.

7 Recommendation

Note for sensor design: The vital part of this project is designing the sensor. If the sensor does not work, nothing will work. So, please find out the designing the sensor and implement this project. There are three test cases which can be reference for the project. And new sensor can be designed for better performance.

We can find out hardware protocol to communicate with computer like RS232 but will communicating with computer application need to do unnecessary extra efforts to establish standard communication protocol beside its objective. So, there is a possibility to develop a standard protocol for in-depended protocol for such device.

8 Annex

8.1 Lhujha Firmware in Assembly Language

Code Sample in Assembly Code

```
,** *****  
,  
,** TITLE : Warrior 2011  
,** DATE : 14 FEBRUARY 2011  
,**  
,** P1.0 : PUMP-1 STATUS  
,** P1.1 : FULL TANK-1  
,** P1.2 : NORMAL TANK-1  
,** P1.3 : RESERVE TANK-1  
,**  
,** P1.4 : PUMP-2 STATUS  
,** P1.5 : FULL TANK-2  
,** P1.6 : NORMAL TANK-3  
,** P1.7 : RESERVE TANK-3  
,**  
,** ACC (A) IS KEY POINT TO MAKE DECISION  
,** *****  
,** DEVICE : ATMEL MICROCONTROLLER 8051 (AT89C51) OR (AT89S51)  
,** AURTHOR : SAHARA FOUNDATION - RAJENDRA MAN BANEPALI  
,** *****  
,**  
,** DESCRIPTION: THE DEVICE IS USED TO AUTOMATE TWO WATER  
PUMPS  
,** *****  
  
$MOD51  
ORG 0000H ;** STARTING FROM 0H  
  
,** TANK STATUS  
_MotorA BIT P1.0 ;** Out  
_WatFullA BIT P1.1 ;** In  
_WaterOnTap BIT P1.2 ;** In  
_WatReserveA BIT P1.3 ;** In  
  
_MotorB Bit P1.4 ;** Out  
_WatFullB BIT P1.5 ;** In  
;_WatNormalB BIT P1.6  
;_WatReserveB BIT P1.7 ;** In
```

```
INIT:
ACALL WELCOME
ACALL NETWORK

START: ;*** START DEVICE...

;*** CHECKING MOTOR ONE
ACALL OperateMoterA
ACALL OperateMoterB

SJMP START

WELCOME:
;*** DISPLAY WELCOME MESSAGE ON LCD
RET

NETWORK:
;*** Serial Communication
RET

OperateMoterA:
;*** Check MotorA/Tank1 for status
JNB _WaterOnTap, OffMotorA ;*** If Water is at tap, it execute next line, if
NOT MotorA SET Off
JB _WatFullA, OffMotorA ;*** Water Tank1 is full then Switch Off MotorA, Else
exe next line
;*** Checking Reserve Indicator is necessary if you
automate only
SETB _MotorA; ;*** Start Motor A
SJMP OutA

OffMotorA:
CLR _MotorA; ;*** Switch Off MotorA

OutA:
RET

OperateMoterB:
;*** Check MotorB/Tank2 for status
JNB _WaterOnTap, OffMotorB ;*** If Water is at tap, it execute next line, if
NOT MotorB SET Off
JB _WatFullB, OffMotorB ;*** Water Tank2 is full then Switch Off MotorB, Else
exe next line
JNB _WatReserveA, OffMotorB ;*** If tank1 is on reserve status, it switch off
```

```
mother B111111
                                ;*** Checking Reserve Indicator is necessary if you
automate only
    SETB_MotorB; ;*** Start Motor B
SJMP OutB

OffMotorB:
CLR_MotorB; ;*** Switch Off MotorB

OutB:
RET

END
```

8.2 Lhujha Firmware in 'C' Language

Code Sample in "C"

```
/** *****
/** TITLE : Warrior 2011
/** DATE : 14 FEBRUARY 2011
/**
/** P1.0 : PUMP-1 STATUS
/** P1.1 : FULL TANK-1
/** P1.2 : RESERVE TANK-1
/**
/** P1.3 : PUMP-2 STATUS
/** P1.4 : FULL TANK-2
/** P1.5 : Water Status On Tap
/**
/** *****
/** DEVICE : ATMEL MICROCONTROLLER 8051 (AT89C51) OR (AT89S51)
/** AUTHOR : SAHARA FOUNDATION - RAJENDRA MAN BANEPALI
/** *****
/**
/** DESCRIPTION: THE DEVICE IS USED TO AUTOMATE TWO WATER
PUMPS
/** *****

#include <reg51.h>

    /**** Water Tank Status
```

```
    //;*** TANK STATUS
    sbit_MotorA      = P1^0;      //;*** Out
    sbit_WatFullA    = P1^1;      //;*** In
    sbit_WatReserveA = P1^2;      //;*** In
    sbit_MotorB      = P1^3;      //;*** Out
    sbit_WatFullB    = P1^4;      //;*** In
    sbit_WaterOnTap  = P1^5;      //;*** In

    //*** Switch State Status
    typedef enum{
        On=1, Off=0
    }TheSwitch;
    //TheSwitch iSwitch;

    //*** Motor Engine Indicator
    typedef enum{
        MotorA=0, MotorB=1
    }Motor;
    //Motor iMotor;

    //*** Function-Prototype
    void DisplayWelcome();
    void SetRS232Network();
    void OperateMotorA();
    void OperateMotorB();
    void SwitchMotor(Motor, TheSwitch);
    //*** Function-Prototype

void main(){
    DisplayWelcome();
    SetRS232Network();
    for (;;) {
        //Programme to check continue...
        OperateMotorA(); //Call Operation of Motor A
        OperateMotorB(); //Call Operation of Motor B
    } //Programme Loop
} //Main Function

void OperateMotorA(){
    //*** If Water On Tap
    if (_WaterOnTap){
        if (!_WatFullA){
            SwitchMotor(MotorA, On);
        }
    }
}
```

```
        }else{
            //Switch Off Motor A since the tank A is Full.
            SwitchMotor(MotorA, Off);
        }//end If Water Tank A is not Full
    }else{
        //Switch Off Motor A since there is no Water on the TAP.
        SwitchMotor(MotorA, Off);
    }//end If Water on Tap
}//Operation of MotorA

void OperateMotorB(){
    /*** If Water On Tap
    if (_WaterOnTap){
        if (!_WatFullB){
            if (!_WatReserveA){
                SwitchMotor(MotorB, On);
            }else{
                //Switch Off Motor A since the tank A is Full.
                SwitchMotor(MotorB, Off);
            }//end If Water Tank A is not Reserved
        }//end If Water Tank B is not Full
    }else{
        //Switch Off Motor A since there is no Water on the TAP.
        SwitchMotor(MotorB, Off);
    }//end If Water on Tap
}//Operation of MotorB

void SwitchMotor(Motor m, TheSwitch s){
    if (m == MotorA){
        _MotorA = s;
    }else if(m == MotorB){
        _MotorB = s;
    }//Motor decision
}//SwitchMotor()

/*** RS232: Serial Communication Section ***/
void SetRS232Network(){
}//SetRS232Network
/*** RS232: Serial Communication Section ***/

/*** LCD Section ***/
/*** LCD: Welcome Message Display
void DisplayWelcome(){
}//DisplayWelcome
```

//*** LCD Section ***/