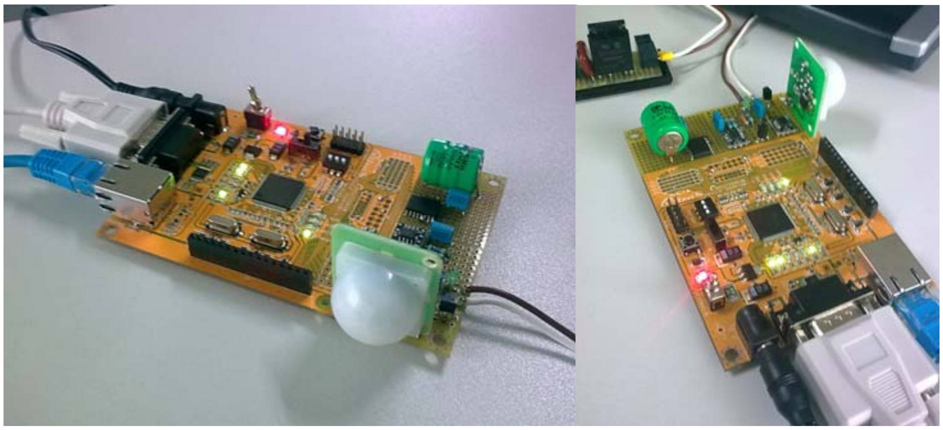
TCP/IP Fuzzy Temperature Controller

Date: June 27, 2010
Project Number: 003199
Eligible part: iMCU7100EVB



The device uses TCP/IP technology and fuzzy logic to control the room temperature based on real temperature, human presence, time of the day and the actual power consumption of similar devices in the building. All this devices are connected in a LAN and exchange information acting both like HTTP client and HTTP server. A web based interface is provided for the device configuration. Two fuzzy control algorithms are used: **temperature fuzzy controller** and **power management fuzzy controller**. They work in cascade and have as the main target maintaining the preselected temperature and consume no more than preselected power threshold.

Project background

Installing electronic temperature controllers for driving the heating system inside my house, made me happy. “Set and forget concept”, provided the user the possibility of creating a temperature controlling plan for the whole week. Everything looked great at the beginning. I considered that I had to set them once, and never touch them again. Than I started using them. It happened a lot of time to:

- live the home during the weekend -> I had to adjust the controllers because the weekend program was set for 22°C and it made no sense to turn the heat on to have this temperature, when nobody was at home
- work until late in the night -> I had to adjust the controllers settings again, because the night program has a lower temperature than the evening program and I was freezing
- be at home a whole week during the holidays or free days -> I had to adjust the controllers again because the daily program between 9:00AM - 17:00PM was set for a lower temperature.

The problem was not only that I had to adjust the controllers all the time according to a given situation in order to have a comfortable temperature, but a lot of electrical energy was wasted because the power consumption was not optimized at all. For example, according to the plan, it was normal for the controllers to turn the heat on, even if nobody was in the rooms.

A totally new type of controller with a totally new concept had to be designed and I started to draw up some requirements for it.

- The temperature controller should take into account the following factors: the actual room temperature; the preselected threshold temperature; the time of the day; the human presence inside the room
- All the power consumption of the heating system inside the house should not exceed a preselected power threshold. This means that the TCs (Temperature Controllers) should adjust their behavior according to the power consumption of the heating elements controlled by the other TCs.
- In order to accomplish the previous requirements the TCs have to exchange information between them.
- A dynamic priority system has to be defined. For example, if the temperature in a room is very low,

the controller in that room will increase its priority. When the temperature will rise above the threshold, the controller will decrease its priority.

- A flexible and easy to use interface has to be provided to the user.

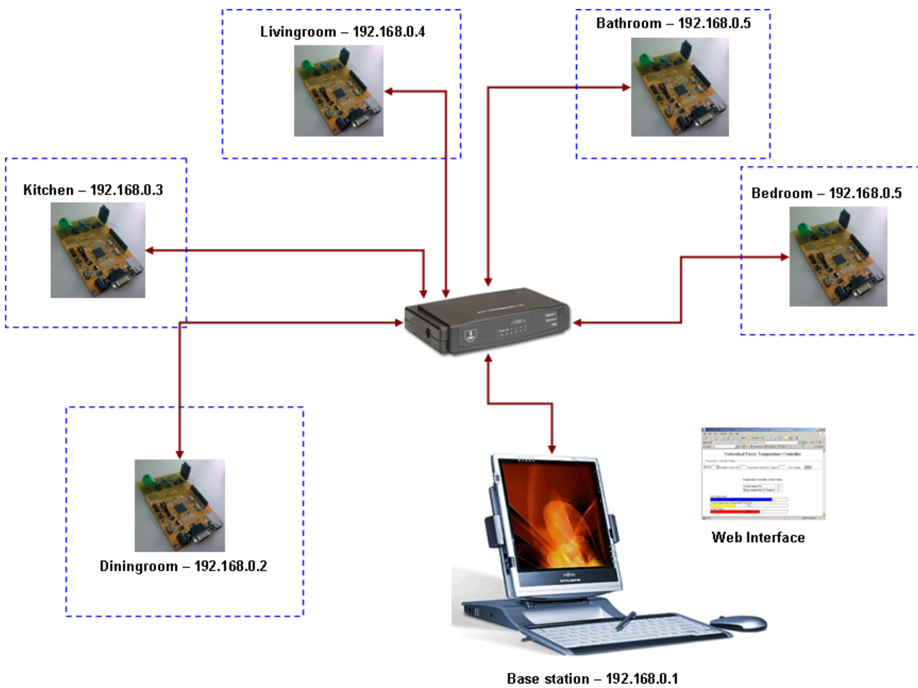
Extracting a behavioral equation to compute the output of the controller in the given condition, would be quite a challenge. Considering that some algorithm can be finally define it has to be also implemented in a powerfull microcontroller and so on. This is the place where fuzzy logic can be applied. It acts according to a set of rules defined by the human expert and saved inside the controller.

Summing up all the presented facts I could extract the following conclusion:

I needed a device (microcontroller) to be able to compute fuzzy algorithms, to offer networking connection possibility, to be able to communicate over standard protocols, to offer an easy to create and easy to use user interface, to have enough RAM and flash memory, and to have a low price. W7100 fulfilled all this requirements.

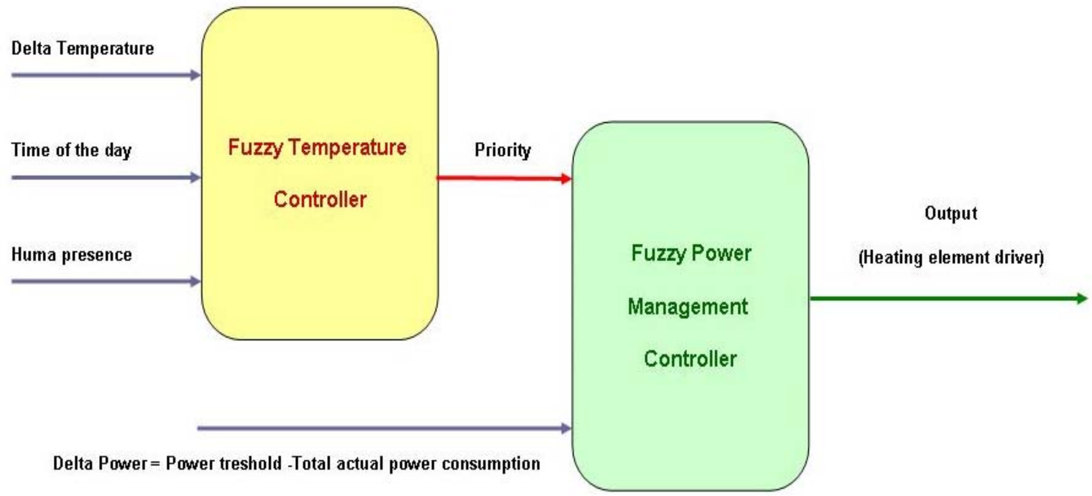
The solution

Considering a TC for every room in the house, we create an Ethernet devices network. The devices can connect to each other and exchange information, acting both as HTTP client and HTTP server. Each device can be accessed from a computer, via a web interface.



Each device has its own priority. This priority changes according to the room condition (temperature, human presence, etc). Prior changing the output, a TC has to check the total power consumption of the other heating systems (TCs and heating elements) in the house. A first fuzzy controller is used to adjust the output according to its own the priority and the total power consumption of the heating system: **Fuzzy Power Management Controller**

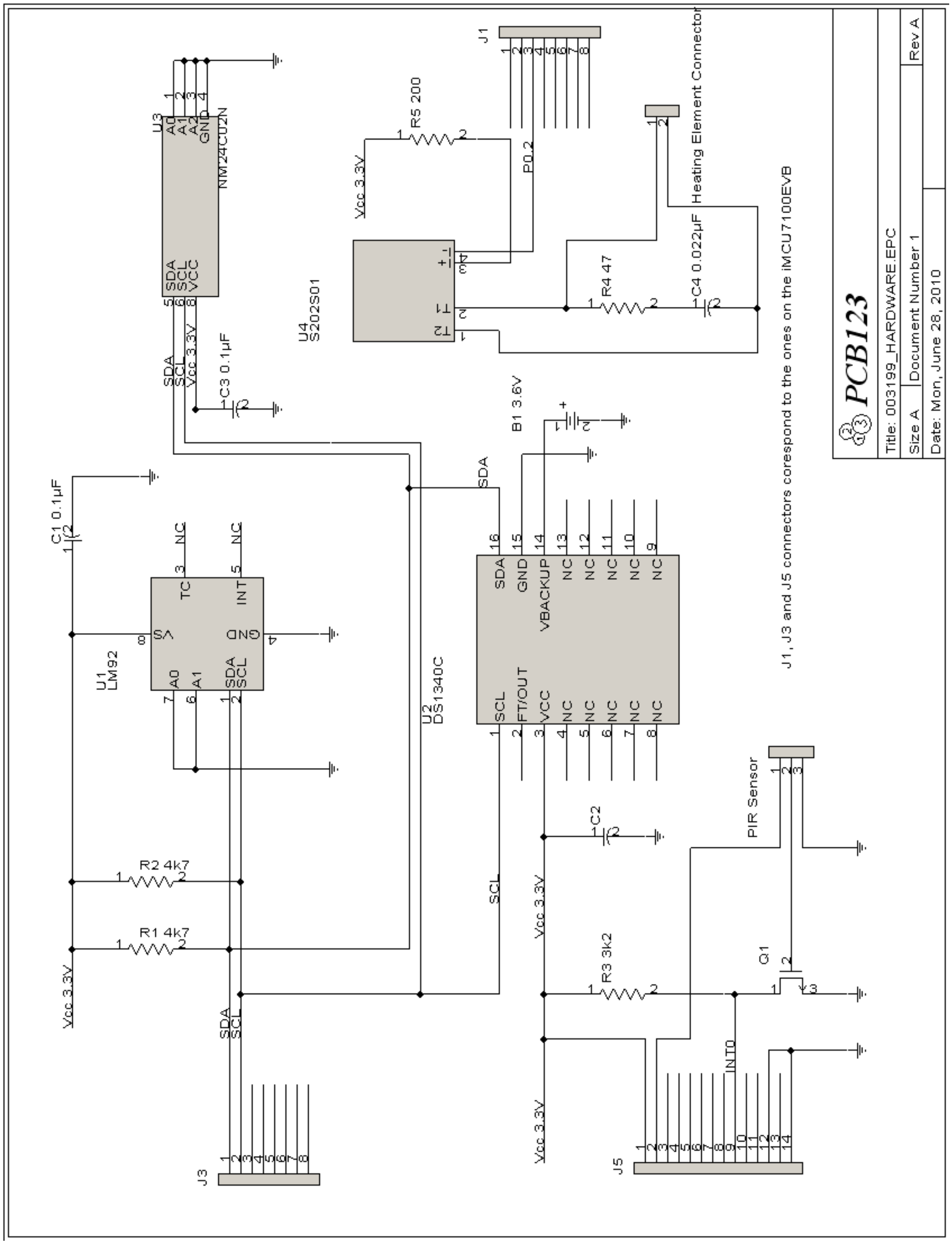
To compute its own priority a TC has to measure the room temperature, to check the preselected temperature threshold, to ask a RTC (Real Time Clock) about the time of the day and check the human presence in the room. This factor represents the inputs of the second fuzzy controller: **Fuzzy Temperature Controller**. The output of this controller will be the **device priority**.



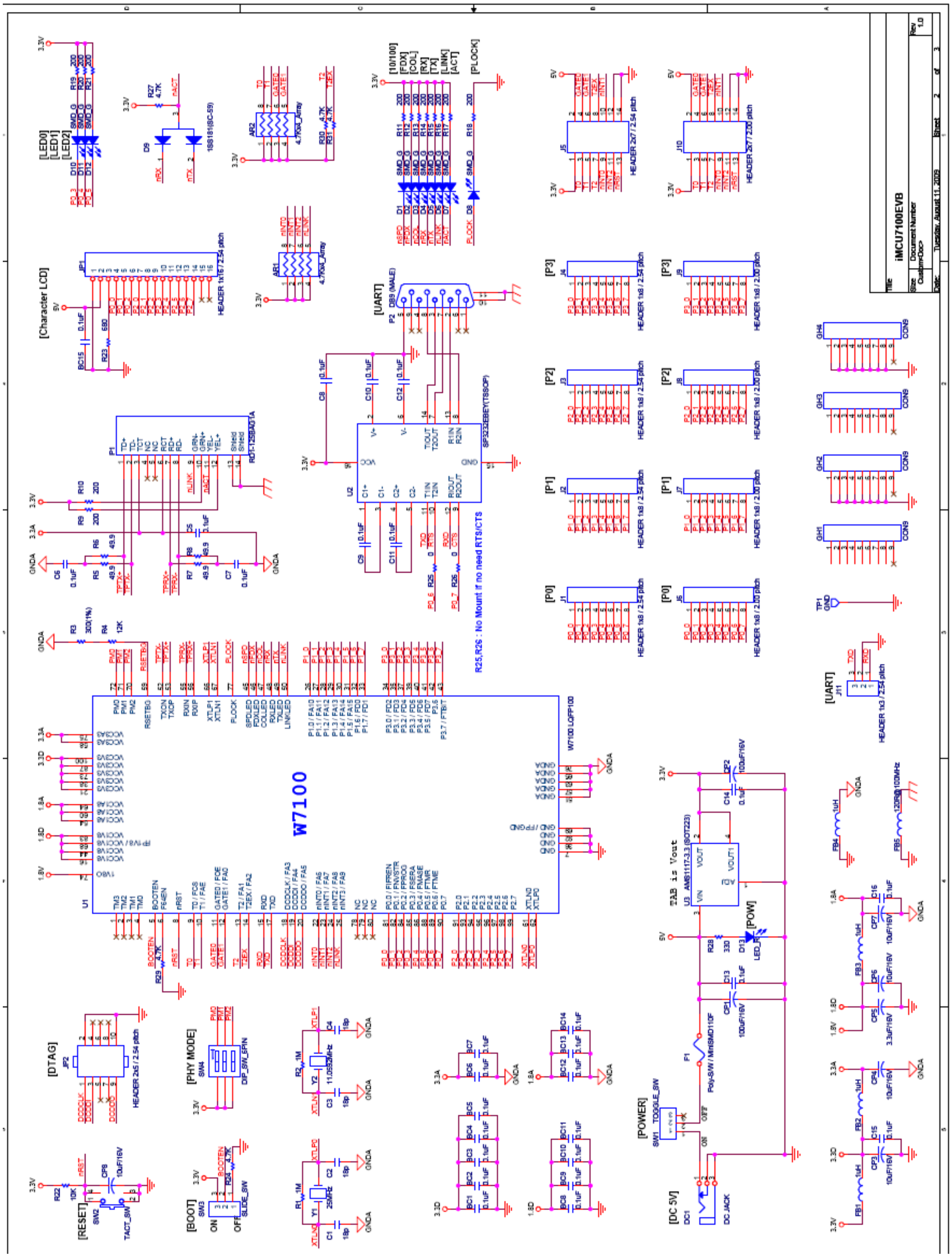
Conclusion

Due to the new available technologies at affordable price, modern concepts like smart house, inter device communication, fuzzy logic, TCP/IP communication, standardized open protocols like HTTP can be embedded and implemented even in a banal temperature controller transforming it in a next generation device.

Electronic schematic of the modules added to the iMCU7100EVB



Electronic schematic of the iMCU7100EVB board



Sample code

```
float power_controller(float delta_p, float priority)
{
        float xdata power[5];
        float xdata prty[4];
        float xdata f[5],m[5];
        int   xdata i,j,n;
        float xdata l_m,r_m,yi,myi,s1,s2,y,r;
        float xdata W[4][5];

        //***********************************************************************
        //fuzzyfication
        //***********************************************************************

        //Delta Power
        power[0] = left_sat(-2.5,-1.5,delta_p);
        power[1] = delta(-2,-1,0,delta_p);
        power[2] = delta(-0.5,0,0.5,delta_p);
        power[3] = delta(0,1,2,delta_p);
        power[4] = right_sat(1.5,2.5,delta_p);

        //Time interval
        prty[0] = left_sat(0,30,priority);
        prty[1] = pi(20,30,50,70,priority);
        prty[2] = delta(50,75,90,priority);
        prty[3] = right_sat(75,100,priority);

        //***********************************************************************
        // logical inferences
        //***********************************************************************

        for(i=0;i<5;i++) f[i] = 0;

        for(i=0;i<4;i++)
        {
                for(j=0;j<5;j++)
                {
                        W[i][j] = min(prty[i],power[j]);

                        switch(FAM_power[i][j])
                        {
                                case 'Z':
                                                if(f[0] <= W[i][j]) f[0] = W[i][j];
                                                break;
                                case 'L':
                                                if(f[1] <= W[i][j]) f[1] = W[i][j];
                                                break;
                                case 'M':       if(f[2] <= W[i][j]) f[2] = W[i][j];
                                                break;
                                case 'H':
                                                if(f[3] <= W[i][j]) f[3] = W[i][j];
                                                break;
                                case 'V':
                                                if(f[4] <= W[i][j]) f[4] = W[i][j];
                                                break;
                                }

                }
        }

  //***********************************************************************
```

```
// Defuzzyfication
//***********************************************************************

        l_m = calc_lmar(f);
         r_m = calc_rmar(f);

        n  = 100;
        s1 = 0;
        s2 = 0;

        for(i=0;i<n;i++)
        {
                yi = l_m + ((r_m - l_m)/n)*i;

                r    = left_sat(0,1,yi);
                m[0] = min(f[0],r);

                r    = delta(0.5,10,30,yi);
                m[1] = min(f[1],r);

                r    = pi(20,30,40,50,yi);
                m[2] = min(f[2],r);

                r    = pi(40,50,60,80,yi);
                m[3] = min(f[3],r);

                r    = right_sat(70,80,yi);
                m[4] = min(f[4],r);

                myi = 0;

                for(j=0;j<5;j++)   if(myi<m[j]) myi = m[j];

                s1 += yi*myi;
                s2 += myi;
        }

        if(s2 ==0)          s2 = 1;

        y = s1/s2;

        return y;

}
```