**Microprocessor based water level controller in**

**domestic water storage tank**

**Students:** Batyr Onashkul, Alisher Lenkhan.

**Group:** IT2-2011

**Description:**

The goal of our project is to create a water sensor that will give a signal if the water level changes and control the speed of the pump, thanks to which the pump will adapt to the situation.

**Statement:**

Every day technologies develop and become more accessible, so we decided to choose a topic that is useful in the economy, not only in the economy can be used in large enterprises. Imagine that people who live in a private house outside the city bought this microprocessor and installed it easily, thereby making their life easier, they no longer have to monitor the water level.

**Methods:**

The standard virtual water level recorder consists of a weatherproof housing that houses the data logger, level sensor and power supply, and comes complete with a solar panel and data shuttle. The system is powered by rechargeable, sealed, maintenance-free batteries with a built-in 12V/10W solar panel that easily keeps batteries charged throughout the year. Extract data with Data Shuttle to your computer. Suitable for installation in various places. Memory range: more than 8250 data sets, expandable to more than 16500.

**References:** <https://www.ukessays.com/essays/engineering/microprocessor-based-water-level-controller.php>

**Status Report**

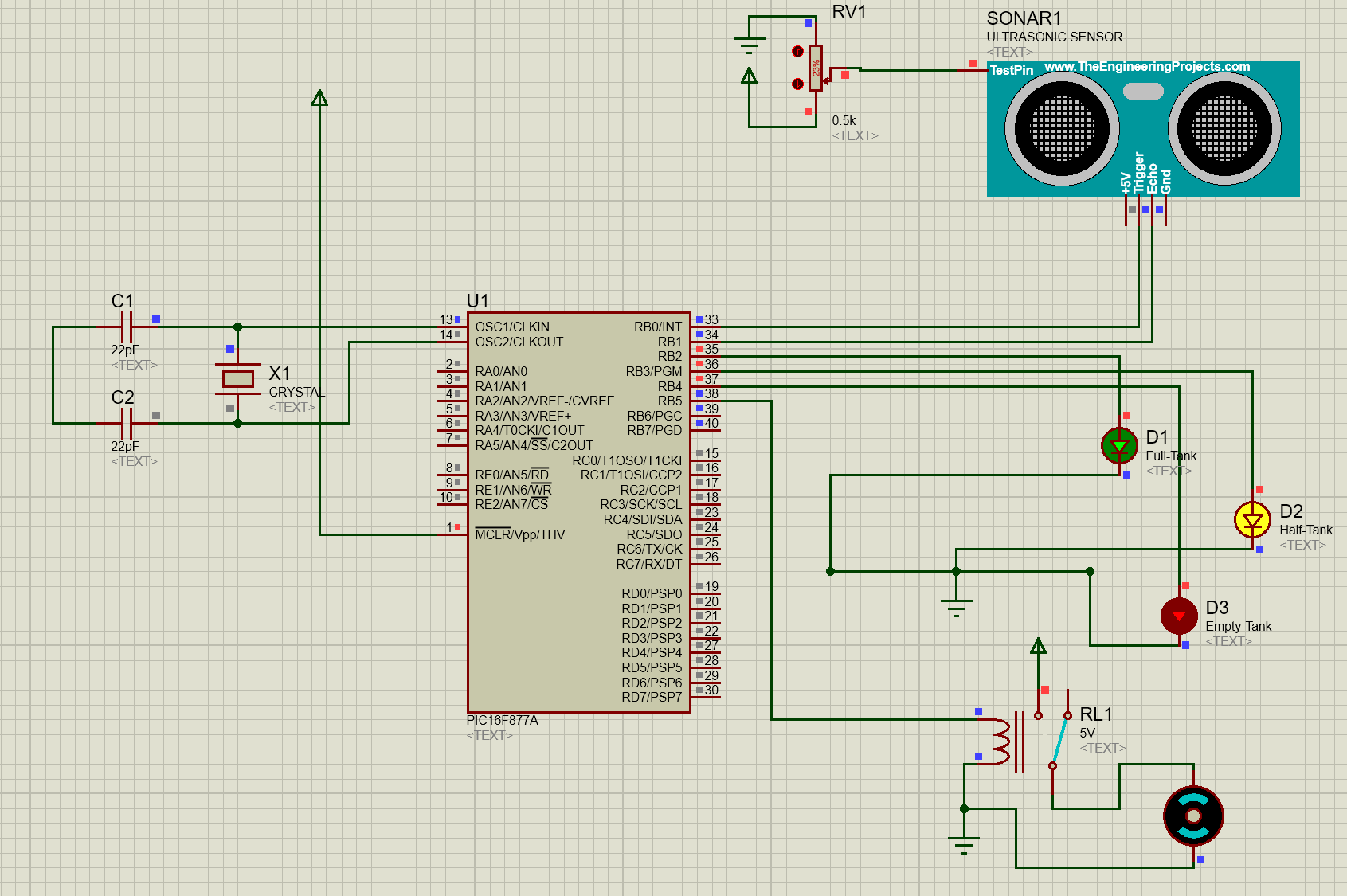
Automated Water Level Monitor System. The main purpose of this device is to provide its users a hassle free experience. Just install the system once and you are good to go. The device monitors the water level inside your reservoir using Ultrasonic/Sonar sensor and then controls the water pump accordingly. Our project at this time is thought out and calculated. We have a schematic, simulation design. The project is at the simulation stage, it works, the code is written.

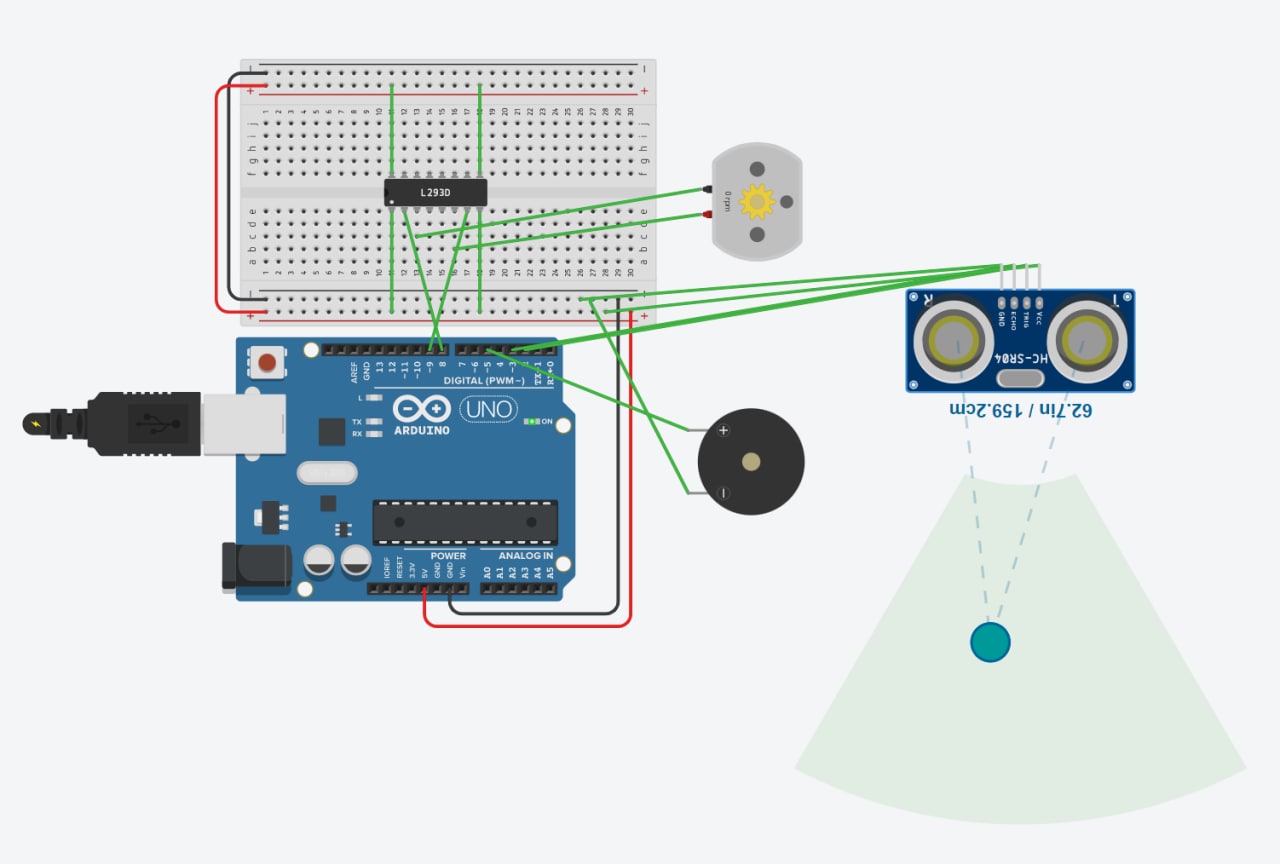
**Progress Report**

Suggested simulation design:



How the design ended up:





All of our moves and procedure how to make this project:

1. We had to set up the distance for our water tank using the Ultrasonic sensor. We had three levels of water tank: full, half and empty. For fixing up the ranges we used the serial monitor of the microcontroller and used code to generate a signal and take the readings.

int distance = 0; 0-10 Tank is empty

int distance = 11; 11-35 is a Medium Level of Water inside a Tank

int distance = 35; 35-40 means that Tank is full

1. We then made our simulation using the Proteus simulation tool.
2. The Sonar sensor’s trig pin is connected to the Pic’s RB0 pin as output and echo pin is in RB1 as input.
3. Then the rest of the pins such as RB2, RB3, RB4, RB5 are connect to green led,yellow led,red led and relay.
4. So when the input comes from the sonar to the pic depending on the condition the LED’s and the relay will light up.

// RB2 is green led pin full

// RB3 is yellow led pin half

// RB4 is red led pin empty

// RB5 is for motor

if(distance < 10){

RB2\_BIT = 0;

RB3\_BIT = 0;

RB4\_BIT = 1;

RB5\_BIT = 1;

while(distance<10){

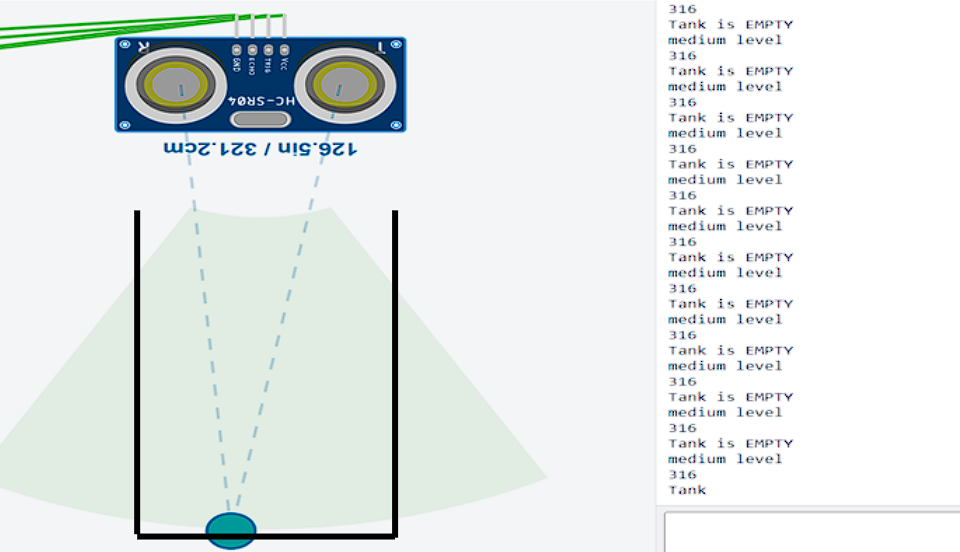
distance++;

delay\_ms(1000);

}

}

1. To see how it works we created simple Ultrasonic Sensor circuit in TinkerCAD where we were able to choose a distance by clicking in the area of the distance. But anyway, in code we were declaring the distance because proteus won’t allow us to select the distance that we want.
2. The Relay will be turned on until the tank gets full of water. When the water tank is full, the relay turns off.



Code for this circuit:

void void main(){

int distance = 0;

TRISB = 0x02;

PORTD = 0x0;

PORTB = 0x0;

TMR1L = 0x0;

RB2\_BIT = 0;

RB3\_BIT = 0;

RB4\_BIT = 0;

RB5\_BIT = 0;

while(1){

if(distance < 10){

RB2\_BIT = 0;

RB3\_BIT = 0;

RB4\_BIT = 1;

RB5\_BIT = 1;

while(distance < 10){

distance++;

delay\_ms(1000);

}

}

else if(distance >= 10 && distance < 35){

RB2\_BIT = 0;

RB3\_BIT = 1;

RB4\_BIT = 1;

RB5\_BIT = 1;

while(distance < 35){

distance++;

delay\_ms(1000);

}

}

else if(distance >= 35 && distance <= 40){

RB2\_BIT = 1;

RB3\_BIT = 1;

RB4\_BIT = 1;

RB5\_BIT = 0;

}

}

// distance function for out Sonar sensor

int calcDistance(){

int temp = 0;

TMR1L = 0x0;

TMR1H = 0x0;

RBO\_BIT = 1;

PBO\_BIT = 0;

T1CON.FO = 1;

T1CON.FO = 0;

temp = (TMR1L | (TMR1H << 8));

temp = (temp \* 0.034)/2;

return temp;

}

}