**Home Water Reservoir Automation System**

**Project Report**

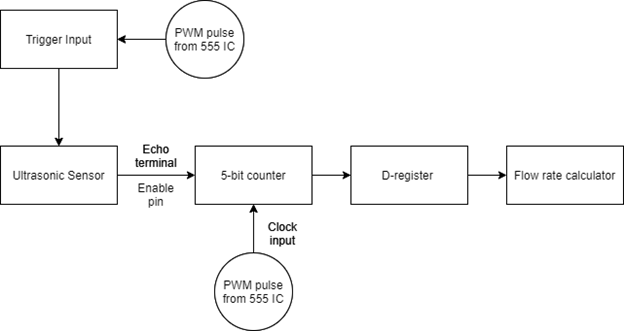
**Team Members:**

1. Ali Hamza Malik CMS: 291480
2. Ali Aqdas CMS: 285050
3. Muhammad Ibrahim CMS: 291898
4. Muneeb Elahi Malik CMS: 288106

**Ultrasonic Sensor Water Level Calculator:**

We have used an ultrasonic sensor to detect the water level. A 555 IC provides a PWM pulse which is used as a trigger input. As soon as the ultrasonic sensor is triggered, the transmitter transmits an 8-pulse wave which is reflected by the water surface and received at the receiver. The echo terminal stays high during the process. So, we used a 5-bit up counter to count the time for which the echo terminal stays high.

**BLOCK DIAGRAM:**



**Sequence of Design steps:**

1. Ultrasonic sensor requires a trigger input to be operational. A 555 IC provides a 10us pulse after which the sensor emits 8 cycle burst of ultrasound at 40 kHz and raise its echo, which then strikes water surface and gets reflected.
2. To calculate the time for which the echo terminal is high, the echo output is connected to the reset input of all 5 JK flipflops serving the purpose of a 5-bit counter.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Clock** | **E** | **D** | **C** | **B** | **A** | **Decimal Value** |
| 0 | X | X | X | X | X | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 2 |
| 1 | 0 | 0 | 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 1 | 0 | 0 | 4 |
| 1 | 0 | 0 | 1 | 0 | 1 | 5 |
| 1 | 0 | 0 | 1 | 1 | 0 | 6 |
| 1 | 0 | 0 | 1 | 1 | 1 | 7 |
| 1 | 0 | 1 | 0 | 0 | 0 | 8 |
| 1 | 0 | 1 | 0 | 0 | 1 | 9 |

**.**

**.**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 0 | 1 | 29 |
| 1 | 1 | 1 | 1 | 1 | 0 | 30 |
| 1 | 1 | 1 | 1 | 1 | 1 | 31 |

1. At the clock input of the counter, a pulse wave of 1330 Hz is given via a 555 IC (assuming the length of the water tank to be 4m, which is crucial to our design because for different depth of water tank a different clock frequency is required). This frequency is set so that the maximum time taken by the echo pin to stay high synchs with exactly 32 counts. So, if the water tank is empty (we have the max distance) it gets 31 counts starting from 0.

**Problems Faced:**

* Synchronizing exact 32 counts with the max time for which the echo terminal could stay high.
* Eliminating the oscillations in LED display

**Flowrate Monitor:**

After getting the input from sensor, the Depth calculator measures the depth of the water at a particular instant of time. It then sends the signals in the form of 8 bits binary number to the flow-rate monitor. The flow rate monitor predominantly stores those values and compares each current value of depth with the preciously stored value. After comparing, it decides whether the water in the tank is flowing out, flowing in or static. The flow rate monitor also, subtracts the previous value from the current value using the 8-bit binary subtractor and yields the absolute value of flow rate at a particular instant of time in clock pulse.

**BLOCK DIAGRAM:**

Diagram

Description automatically generated

**Sequence of Design steps:**

Following are the sequence steps followed during the design of flow rate monitor.

1. The first thing that flow rate monitor does is store the 5-bit binary value coming from the depth calculator as soon as the system is turned on. Hence, we accomplish this task by using a D-register (IC 74273) with a clock input and master reset pin.

|  |  |  |  |
| --- | --- | --- | --- |
| **D Input (8-bits)** | **MR** | **CLK** | **Q** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Q(t)** | **D** | **CLK** | **Q(t+1)** |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

1. The next step in our design is to implement the procedure for comparing the current value from the depth calculator and the previously stored value in the D-register. For this purpose, we use an 8-bit magnitude comparator. In proteus we used two 4-bit binary comparator ICs (7485) in such a way that they implement the logic for 8-bit magnitude comparator because the 8-bit magnitude comparator has no simulator profile in the proteus. We use 5-bit of the logic IC’s for our purposes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Q(t)** | **Q(t+1)** | **Outputs** | | |
| **Q(t) >Q(t+1)**  **Flow out** | **Q(t) = Q(t+1)**  **No Flow** | **Q(t) < Q(t+1)**  **Flow in** |
| 00000000 | 00000000 | 0 | 1 | 0 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| 00001000 | 00010000 | 0 | 0 | 1 |
| . | . | . | . | . |
| . | . | . | . | . |
| . | . | . | . | . |
| 01000000 | 00010000 | 1 | 0 | 0 |
| . | . | . | . | . |

1. Our next task is to find the difference of current value of depth and previously stored value for depth at an instant of time to find the value of flow rate. For this purpose, we use an 8-bit binary subtractor. We implement the 8-bit binary subtractor using two full adders (IC 7483) and a combination of XOR gates (IC 7486) at the input of those adders. The carry in of the first full adder is set to permanent high input in order to achieve the logic of subtractor.

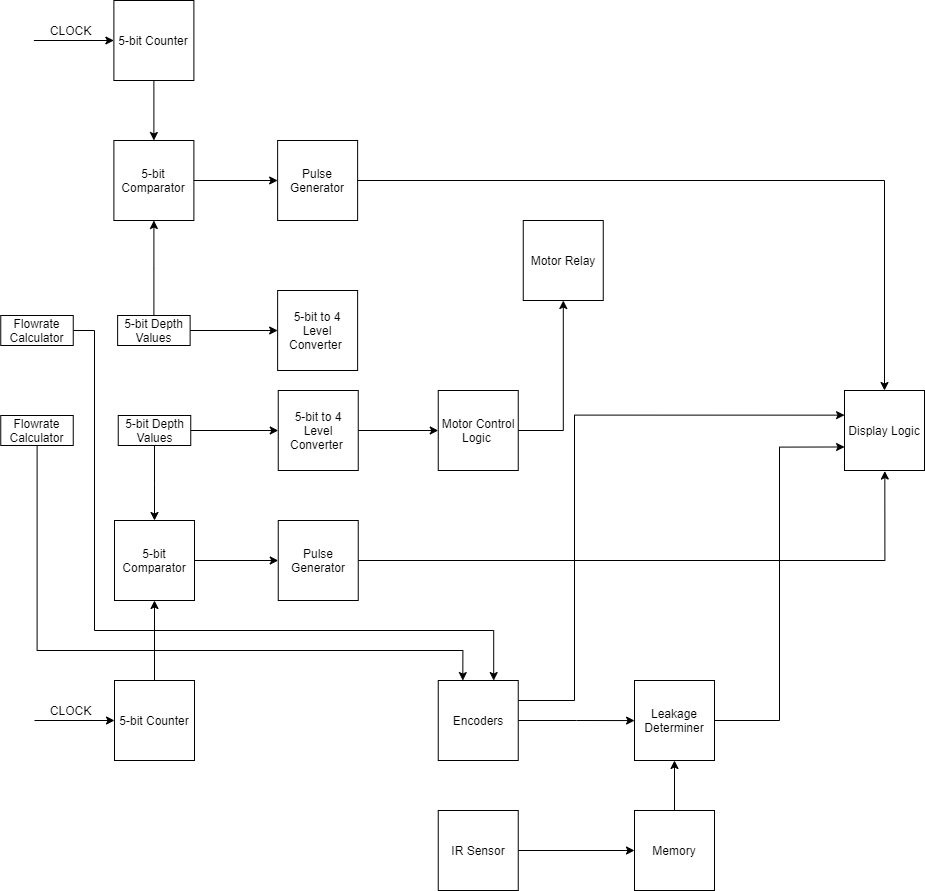
**Problems Faced:**

* The 8-bit magnitude comparator was not present in the proteus; therefore, I had to use two four-bit magnitude comparators and implement the logic of 8-bit comparator with three outputs.
* Similarly, the binary subtractor was not available in proteus and I used two full adder ICs with an arrangement of XOR gates at their inputs to implement the logic of Binary subtractor.

**Control Logic:**

The Depth Calculator gives depth value encoded as 5-bit binary value using an Ultrasonic Sensor where binary 0 is empty reservoir and binary 31 is full reservoir. This 5-bit depth value is fed through a flow calculator which gives a 5-bit value of flow rate and determines if flow rate is positive or negative. These values are fed to the control logic which converts the binary valid to pulses and count mode which are fed through two wires to the display which displays water levels as led bar levels. The depth level values are converted into 4 levels represented by led bars which are fed to the motor control logic which decides when to turn motor on or off. The Control Logic also takes value from flow rate calculator and the Infrared sensor logic to determine if a leak is occurring or not.

**BLOCK DIAGRAM:**



**Sequence of Design steps:**

Following are the sequence steps followed during the design of flow rate monitor.

1. The first step was designing the logic to generate, increment and decrement logic pulses using a 5-bit comparator and 5-bit counter. The purpose of this is to transfer information from control logic to the display module with minimum number of wires.
2. The Enable pin of counter is connected to A=B pin of comparator such that when (A=B)=1, the counter stops counting and when (A<B)=1, it counts up and when (A<B)=0 and (A=B)=0, it counts down. The Enable pin is also connected to an AND gate with other pin connected to the CLOCK to generate Pulses to be passed to Display Logic. The Number of pulses is equal to increment and decrement of the water levels and the mode pins determines if level rises or falls.
3. In order to adjust the sensitivity of motor control for different dimensions of water reservoir, the 5-bit binary water level value is converted into 4 level values which are feed to an SR latch through AND gates such that when both water levels reach certain value, S goes high and when Overhead Reservoir is full, R goes high and Q is connected to Motor through an AND gate and other terminal is connected to Empty detector output for Underground Reservoir Empty detector through not gate. The Water depth to 4 level value truth tables are given below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Overhead Reservoir** | | | | |
| **Water Level Value (ABCDE)**  **(5-bits)** | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| 00XXX | 1 | 1 | 1 | 1 |
| 01XXX | 0 | 1 | 1 | 1 |
| 10XXX | 0 | 0 | 1 | 1 |
| 11XXX | 0 | 0 | 0 | 1 |
| WO | (A+B)’ | A’ | (AB)’ | 1 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ground Reservoir** | | | | |
| **Water Level Value (ABCDE)**  **(5-bits)** | **Level 1** | **Level 2** | **Level 3** | **Level 4** |
| 00XXX | 0 | 0 | 0 | 1 |
| 01XXX | 0 | 0 | 1 | 1 |
| 10XXX | 0 | 1 | 1 | 1 |
| 11XXX | 1 | 1 | 1 | 1 |
| WG | AB | A | A+B | 1 |

1. The level functions are multiplexed into a single output and the selection line is connected to a counter with clock pin connected to a push button to switch between levels.
2. In order to detect if there is leakage in water pipes, we need to know if there is someone in the Washrooms/Kitchen and if water is being used. We do this by placing IR sensors in parallel and wiring them up to a circuit based around JK Flip Flop.

|  |  |  |
| --- | --- | --- |
| **IR1** | **IR2** | **Q(t+1)** |
| 0 | 0 | Q(t) |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | Q’(t) |

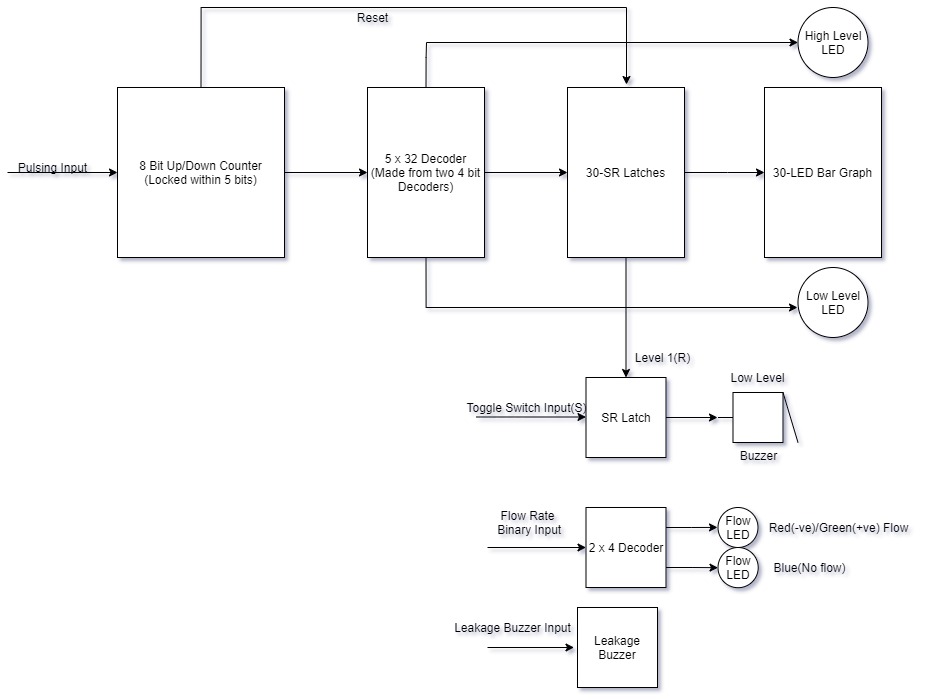
**Problems Faced:**

* A startup issue due to simulation not running in real time caused a lag in modules which was fixed by adding Reset pins to all modules to which a counter is connected that resets all modules for a few clock cycles of trigger pin of ultrasonic sensor.
* 5-bit counters, comparators, adder/subtractor etc. were not available or simulation model was not present so we setup sub-circuits for the ICs.

**Display and Warning System:**

The data received from the control system to the display system for water level detection is in the form of pulses. Whenever the water level increases the control systems send a pulse and same happens when the water level decreases. Therefore, the data can be transmitted only with two lines instead of 5 lines for 5 bits. The two lines from the control system can then be used as the input of the clock signal to an Up/Down counter.

**BLOCK DIAGRAM:**



**Sequence of Design steps:**

The design process of the display system is as follows

1. The first step is the implementation of a 5 bit Up/Down counter from a 4 bit Up/Down counter which is accomplished by using two 4 bit Up/Down Counters and a combinational logic that does not let the counter count above 5 bits and in negative. Truth Table for the combinational logics is given below. We require **CLKU** pin to be **1** whenever **I** is **0** or **Q** is **1.** This will inhibit it from counting over 5 bits and reset to zero after all five bits are **1**.

|  |  |  |
| --- | --- | --- |
|  |  | **CLKU** |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Similarly, to avoid the counting in negative we have used the pin of counter. From observation it is seen that whenever the counter is zero and is switched to low the borrow bit becomes 0. And when becomes 1 after it becomes 1. Thus a positive edge occurs. XOR of and a reset input from control system is the attached to clear pin which resets the counter. This behaviour is shown below in a compact table which shows variables of concern.**.**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 0 | 0 | 1 |

|  |  |  |
| --- | --- | --- |
|  |  | CLEAR |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| is remains 1 as long as counter doesn’t decrement to a negative number. needs to be kept high to maintain 1 at **CLEAR** pin. Now if becomes zero and then becomes 1 a positive edge occurs at CLEAR pin. Similarly if switched to 0 and then back to 1 a positive edge occurs. To keep initially, it is inverted. | | |

1. The next step is the implementation of 5-bit decoder. Since we have 4-bit decoder available in Proteus we implement a 5-bit decoder by controlling it's pins. The fifth bit is attached to pin of first decoder directly and to the second decoder through a **NOT** gate.
2. Now that we have implemented the decoder, we have the store it's output to display the output on the led bar display in an aesthetic manner. This is accomplished by implementing a SR Latch as shown in the image below. Multiple latches are stacked together to form a component that has ten of these latches to store data for LED Bar Graph.

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | Undefined |

1. Now we can connect the output of these latches to the input pin of LED Bar Display. The display will show the water levels but turning on LEDs. Higher number of LEDs turned on means higher water levels and vice versa. Two LEDs other than the led bar display are also attached to display full level and low level warning.
2. Another important component of the system is a buzzer warning system for low levels of water in underground water reservoir. The buzzer rings when the water level is low but it can be switched off by a momentary pulse. The pulse is provided by a button across a high input. The switch is accident proof. If someone turns off the switch the buzzer will automatically turn on next time the water is filled, and then falls to a low level. Below is a state table of the buzzer’s sequential logic.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Current State of Latch** | **SR Latch Inputs** | | **Low Level Warning Input** | **Next State of Latch** | **Buzzer Input** |
|  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | Undefined | X |
| 0 | 1 | 1 | 1 | Undefined | X |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 | 1 | 1 |
| 1 | 1 | 1 | 0 | Undefined | X |
| 1 | 1 | 1 | 1 | Undefined | X |

From table it can be seen clearly that

Furthermore, the states which are not supposed to occur are highlighted in red. The two inputs S and R cannot be 1 at the same time. Furthermore, S and W cannot be 1 at the same time.

1. One other feature is the flow rate display which is provided by multi coloured LEDs. Two Bits of data are provided at input and passed through a decoder to turn on different colored LEDs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **A** | **B** |  |  |  |  |
| 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 |

**Problems Faced:**

SR Latches were not available as required; therefore, I had to make my own latches and convert them to a component. Making a component from gates was a tedious task with very little information available on internet.

**Future Improvements:**

* In most Houses, there are water faucets and valves which can trigger the water leakage system and cause alarm to ring which can be fixed by installing flow sensors to the few water faucets outside washrooms and kitchens.
* A way to adjust the sensitivity of the ultrasonic sensor counter to set the depth calculator for various heights of water reservoir.
* For final version, we need to position display module in a place where it is visible and audible to the residents while Control Box is placed at a place where sensor wires are easily connected up to Control Box. For this, a Serial Method to transfer data over fewer wires would be more efficient and would have lesser cluster of wires