

CAPSTONE PROJECT

I) Reflection

The way I executed the final was by carefully planning out my weekend, I started with the code; since the code offered in Projects 5 6 and 7 were more than sufficient starter code; I just needed to learn the code first for the Relay so it can send a signal to the water pump, and then I found a neat forum page that taught me who to execute the flow meter code(<https://projecthub.arduino.cc/sheekar/yf-s201-water-flow-sensor-and-arduino-sheekar-banerjee-78ea88>) . The application of the values was maintained relatively the same with a few exceptions. There weren't many unexpected issues, I just thought the ultra-sonic sensor was a little too precise for its own good, as water sometimes water fluctuates, so as soon as it hit the water level it needed to start the pump it would go; so I think for future applications give it an error rate let's say 0.5 cm each side; so the pump does not turn on by accident. I would take a second semester of IOT, maybe focusing more on actuators, and maybe building some of our own via 3D printing, maybe incorporate more soldering on the microcontrollers onto builds, etc. My favorite part of the course was probably just the entire Arduino aspect and being able to build things from the ground up, I also don't think I had a least favorite topic, the class had a wonderful blend of hardware and software topics. I think obviously the cost for materials is a big factor in the scope of the class however personally I don't mind it because I had an excuse to buy the full Arduino set, now I am able to develop an engineering portfolio and it opened new possibilities that were not possible hadn't I bought the materials.

```

#define TRIG 7
#define ECHO 8
#define RELAY 5
#define FLOW 2

//From the spec sheet part this pumps Frequency: F=98 * Q (L / Min), error: ± 2%
const float PULSES_PER_LITER = 5888.0;

//distance from the sensor and when the pump and the volume you can't dispensed
(simulating pouring yourself a cup of water)
const float START_DISTANCE_CM = 30.0;
const float TARGET_VOLUME_L = 1.0;

bool PumpRunning = false;

volatile unsigned long flowPulses = 0;
unsigned long LastFlowtime = 0;
float currentFlow_Lmin = 0.0;
float Total_Volume_L = 0.0;
float cycle_startVolume = 0.0;
float level_cm = 999.0;

```

These are the variables global variables used for the pump, sensor, relay and flow meter. Start distance refers to the distance between the water level and the sensor, once the water level hits that start distance it's an indicator for the pump to activate. Cycle_start_Volume refers to the volume it had dispensed prior to the new cycle meaning if it dispensed .5 liters in one cycle, for the next cycle it starts at .5 Liters, and it dispense 1.5, then for 2 cycles it has dispensed 2 liters.

```

/**
 * Measurement types for event trigger
 */
#define MEAS_LEVEL 1
#define MEAS_FLOW 2
#define MEAS_TOTAL 3

```

Here are the measurement types for each trigger, like the DHT SENSOR measurement trigger for temperature and humidity.

```
f (measurement.equals("level")) { //set the measurement sensor
    contactMeasurement= MEAS_LEVEL;
} else if (measurement.equals("flow")) {
    contactMeasurement= MEAS_FLOW;
} else if (measurement.equals("total")){
    contactMeasurement= MEAS_TOTAL;
}
}
```

Here the contact measurements are meant for the water level, the flow of the water and the total volume it dispensed after every cycle. If the measurement is equal to the “level”, then the measurement will go under the level category, subsequently the same for “flow” and “total”

FUNCTIONS FOR SENSORS AND ACTUATORS:

```
float readDistancecm(){
    digitalWrite(TRIG,LOW);
    delayMicroseconds(2);

    digitalWrite(TRIG,HIGH); //produces waves for Echo to receive
    delayMicroseconds(10);

    digitalWrite(TRIG,LOW);

    long duration = pulseIn(ECHO,HIGH,30000); //this is where it receives the waves
    from trig
    if(duration == 0) return 999.0; //no water present
    return duration / 58.0;
}
```

Here is the first function which takes care of the ultrasonic sensor and reads the water level; first it sets the trig pin up so it can start producing waves, when it hits HIGH waves are being produced, and then after switching to LOW for TRIG, then echo is ready to receive the signals as it get turned on (HIGH).

```

void updateFlow(){
  unsigned long now = millis();
  unsigned long delta_t_ms = now - LastFlowtime;
  if(delta_t_ms < 200) return;

  noInterrupts();
  unsigned long pulses = flowPulses;
  flowPulses = 0;
  interrupts();

  LastFlowtime = now;
  if(delta_t_ms == 0) return;

  float pulsesPerSecond = (float)pulses * 1000.0 / float(delta_t_ms);

  currentFlow_Lmin = pulsesPerSecond / 98.0;

  float deltaL = float(pulses) / PULSES_PER_LITER;

  Total_Volume_L += deltaL;
}

```

This function updates the flow of the water in real time. Using <https://projecthub.arduino.cc/sheekar/yf-s201-water-flow-sensor-and-arduino-sheekar-banerjee-78ea88> as the base for the code; every time the water flows inside the flow meter, a wheel is pushed; that wheel generates pulses that are then converted to the flow rate(L/min) and the total volume dispensed (L). The sensor updates in ISR (processor that handles in real time events, like collecting data, by halting the main program in tenths of a second while reading the data). It copies the pulse, resets the pulse reader, then it interrupts as a form of re-enabling; meaning it reads the pulse for every time interval.

FSM and loop function(with drawings in presentation):

INIT: initializes the process and gets the initial sensor values

GET_REQUES: Sees if there is a Python script that requests the data

PARSE_REQUEST: it stores the registration settings

SENSE: it reads the flow level, the water level, and the Volume dispensed by the control pump.

THINK: here it decides whether the data should be sent, what type of data should be sent, and the values.

ACT: based on event, weather rising, falling, or non, and depending in the value whether its flow rate, total volume or distance it builds a message for the serial monitor:

SEND_RESPONSE: it takes that message and values and sends it to the Python Script

Then it loops back to SENSE and repeats until the timer runs out or disconnect the Arduino.

New Features and thought Process:

With the new opportunity of the directed study I want to talk to the Arduino and send it instruction via the website; by inputting the value of the desired amount of water into the website, it sends that value to the Arduino and commences that process that way you can have water dispensed whenever you want instead of it being restricted to just when it hits the water level desired. Probably make the system more waterproof and be able to bring portably anywhere and be adaptable to any “cisterna”.

I thoroughly enjoyed working with networks as well, the idea of sending useful sensor data in real time is very interesting and being able to work upon the basics and be able to have the water flow, total Volume and distance working in real time or near real time would be neat as well. Those are just some of the things that I plan to address through my directed study.

DEMO:

<https://docs.google.com/videos/d/1102aJQ0URJqMCBUgQ6VONVeyg4mXnNPZ6dioE8WSnWk/edit?usp=sharing>

Sensors used:

I) Parts and Resources

a. 48-1 Keye studio Ultra Sonic Sensor

- i. The sensor has strong directivity and low energy consumption. It's a sensor largely used to measure distance with it being used as a range finder and positioning measuring instrument
- ii. It has 4 pins, with two being the Power supply and ground, and the other 2 are Trigger and Echo; these two pins are connected to separate Digital I/O pins.
- iii. The trigger pin sends a 40Hz (naked to the human ear) then measures the distance it takes for the wave to travel back to the Echo pin.
- iv. The longer the echo pulse the more range the sensor can detect since the transmission lasts longer.

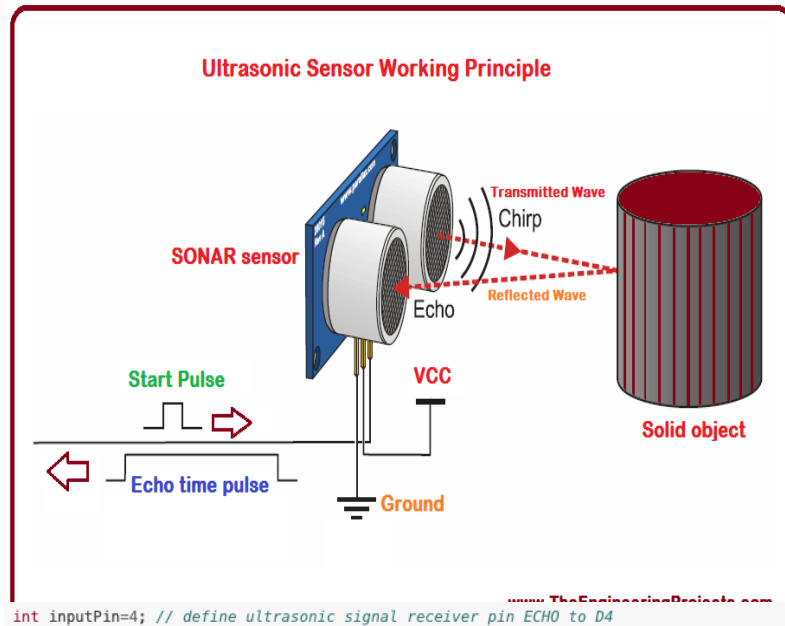
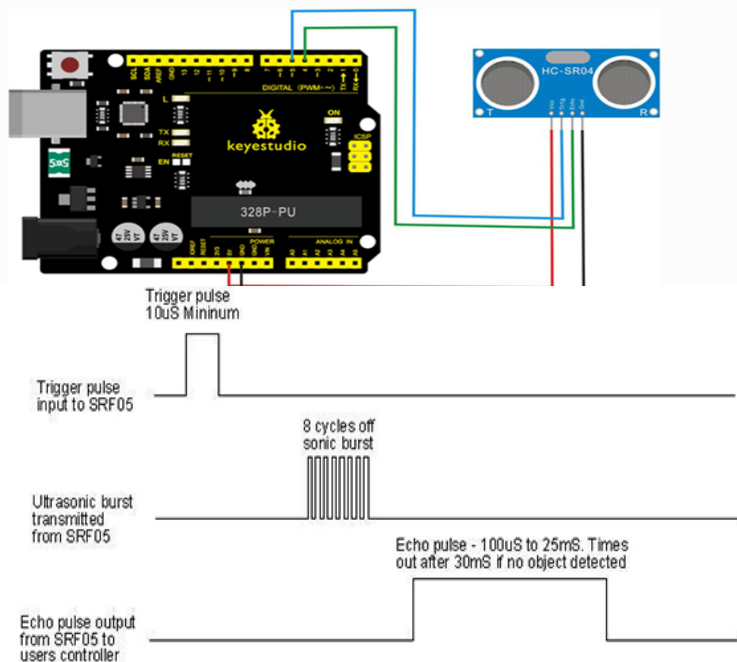
<https://docs.keyestudio.com/projects/KS0349/en/latest/KS0349.html#project-46-ultrasonic-ranger>

- V4.0 board*1
- Ultrasonic sensor*1
- USB Cable*1
- Jumper wire*4

Next, please refer to the following connection table:

Ultrasonic ranger	Arduino Uno
ECHO	D4
TRIG	D5
VCC	5V
GND	GND

Note: D4、D5 are the digital pin 4 and pin 5. You can refer to the connection diagram shown below:



```
int inputPin=4; // define ultrasonic signal receiver pin ECHO to D4
int outputPin=5; // define ultrasonic signal transmitter pin TRIG to D5
void setup()
{
  Serial.begin(9600);
  pinMode(inputPin, INPUT);
  pinMode(outputPin, OUTPUT);
}
void loop()
{
  digitalWrite(outputPin, LOW); delayMicroseconds(2);
  digitalWrite(outputPin, HIGH); // Pulse for 10u s to trigger ultrasonic detection
  delayMicroseconds(10);
  digitalWrite(outputPin, LOW);
  int distance = pulseIn(inputPin, HIGH); // Read receiver pulse time
  distance= distance/58; // Transform pulse time to distance
  Serial.println(distance); //Output distance
  delay(50);
}
```

b. DIGITEN G1/4" Water Flow Sensor (Hall effect Sensor)

i. Frequency: $F=98 * Q$ (L / Min), error: $\pm 2\%$

Flow range:0.3-6L/min

Pressure range: $\leq 0.8\text{MPa}$

ii. Maximum Consumption current:10 mA (DC 5V)

Working voltage range: DC 5-18 V

Operating Temp: 0-80 °C

iii. Operating humidity: 35%-90%RH

Storage Temperature: -25-80 °C

Storage humidity: 25%-95%RH

Output Waveform: Square Wave, output pulse singal.

Cable length:15cm.

iv. Application: water heater thermostat, water purifier, boiler, water dispensers, coffee machines, smart card equipment, the boiler and so on.



ACTUATORS:

a. A Single Relay

- i. A relay is an electromechanical switch that uses a control signal to open or close high-power circuit.
- ii. This single relay was connected between a 9V battery pack and the water pump, since the water pump has a working voltage range of 6-12 Volst and external battery had to be used in conjunction with the relay
- iii. It only needs a singular digital I/O pin, with the COM (common connection, meaning the positive end of the battery is connected to this part) the connection between the battery and pump, and then the pump is connected to Normally Open meaning the circuit is off until the relay is activated.

```
////////////////////////////////////  
int Relay = 8;  
void setup()  
{  
  pinMode(13, OUTPUT);      //Set Pin13 as output  
  digitalWrite(13, HIGH);   //Set Pin13 High  
  pinMode(Relay, OUTPUT);   //Set Pin3 as output  
}  
void loop()  
{  
  digitalWrite(Relay, HIGH); //Turn off relay  
  delay(2000);  
  digitalWrite(Relay, LOW);  //Turn on relay  
  delay(2000);  
}  
////////////////////////////////////
```



b. Gikfun DC 6V-12V Self-Priming Diaphragm Pump - Micro Water Pump

- i. Working voltage: DC 6-12V, Power: less than 6W, Maximum flow: 1.5-1.8L/Min
- ii. Power cord not included. Pump head: 5 meters, pump suction: 1.5 meters (test with water)
- iii. Micro self-priming pumps are widely used in home appliances, medical care, models, aquarium equipment, etc., mainly to achieve functions such as pumping water and cold-water circulation.
- iv. The pump provides a smooth flow over all operating ranges. It also has a self-priming function that allows it to work quietly in the application. The pump can run idling for 30 minutes. When we are doing destructive work, the continuous pumping time can reach more than 1000 hours



[Click to see full view](#)

II) 4 Person Narrative

- a. State of Art: Many of the applications that I have been able to find online are more DIY approaches or the other spectrum which are expensive water sensors in which the price of the instrument far exceeds the building costs, some use actual water sensor instead of ultrasonic sensors, while other use a much more powerful servo motor.
- b. Who cares? The type of person that was chosen for this project was the jibaro. A person from the countryside with little knowledge of the precarious situation the PRASA finds themselves in with water shortages all over the island. They are in desperate need of someone out there to help them with the management of water, and allocation of resources in general. This is especially true after storms or natural disasters where people don't know the next time, they will get water for their house and need to strategically save water.
- c. New Approach? With the Arduino, electrical bills aren't severely affected by any significant power usage, they allow for customization, maybe different Arduinos for different tanks. For drinkable water I found this instructible that allows for you to build a water quality monitor, and you could combine that with this design to have a tank solely for drinking purposes.
- d. Impact: This project has the potential to bring attention to something largely ignored by the mass population of the island, it raises concerns over how safe the water in Puerto Rico is for the less fortunate and offers solutions to a problem that thousands of people face every day.

