

# SOLVING PROBLEMS THROUGH THE INTERNET OF THINGS

Relevant for: Science & Technology | Topic: Science and Technology- developments and their applications and effects in everyday life

Ideas are created from problems | Photo Credit: Getty Images

In this day and age, one is so connected to machines that most of our day-to-day chores are automated. These enhanced capabilities have an overall positive effect as they give us more freedom and opportunity to multitask and be productive while making good decisions. All you need to do is “THINK”. Think of an idea for a problem you want to solve and most of your daily problems can easily be automated and solved. But how do you think of a creative idea? How do you know that the solution will work for you?

In this article, we will go through some ideas on how to start thinking about solutions to everyday problems. The article will help you with some code snippets to get you started, but at the end, it is you who will bring the idea to life. Before we start, let us take a look at the different stages of innovation. It is important to know what we are trying to achieve and where to stop if we want our solution to materialise.

Ideas are created from problems. To make a difference in the world, we need to be able to see the pain points and find a way to solve them. So, do you have a problem that you want to solve now? If not, for the sake of explanation, let us take a common household problem as an example.

There are two water tanks on two floors of a building. The same water pump fills the water in these two tanks. Most of the time there is a delay between checking the optimum water level in the tanks at the two floors and then switching off the water pump because of which there is either water overflowing or tanks not filling completely.

So, how can we solve this problem? Can we think of some mechanism where there is an alarm or notification that one can receive so that the pump can be switched off in time? This will ensure that the tanks are full and no water is getting wasted. Let's start with this idea and visualise it.

To make an idea stick, we need to be able to communicate it in a way that is memorable and understandable. Visualising your ideas can help you do just that. The more you visualise the idea, the more you figure out pointers that might help you in developing the solution. Visualising your ideas can be done through many different mediums such as images, videos, or infographics. These will allow you to break down complicated concepts into smaller pieces so that they are easier to understand.

So, let's try to visualise our above idea for the water pump problem. To start visualising your idea, ask yourself different questions, like “Should I buy an alarm or can an SMS be good enough?”, “How much money do I have to create this solution?” etc. You can also think about what the solution may look like.

Before you start developing the solution, there is just one more step. It is important to figure out the possibilities of extending an idea. Can the idea evolve to solve a bigger problem? If yes, then you have to make your code scalable as well. You don't necessarily have to solve the extended

possibilities right away but while solutionising, keep the door open to scale later. For instance, if you go shopping and you are sure that you are only going to buy three products, you select a shopping bag which can hold just the three pre-determined products. But you could also choose a bigger bag, just in case you find something useful that is not planned.

A lot of people are scared that if they extend an idea, it will get too complicated and that they will lose the original idea. However, this is not true. If you take the time to think about the problem and break it down into smaller parts, you can definitely find a solution to a bigger problem.

For the above example, if we start thinking at a larger scale many possibilities arise; Do we really need to switch off the water pump ourselves? We can maybe extend the solution to automatically switch off the water pump, when the water reaches optimum level. We could also create a mobile app to stop the water pump from anywhere in the world.

Now that we are aware of some of the ways in which the idea can be extended to make our life even better, let's start by creating a solution for our initial problem.

Let's start with the solution now. We might think that to start, we need to directly go and tinker with the water pump and the water tank. Well the good news is that you don't have to. We can start creating the solution directly from our desks.

When we try to create a software program to solve big problems, we use a concept called Stubbed Data. This means using a fake/simple set of actual data that you can feed to your programme to check if it is working fine and to make necessary changes based on the feedback you receive. After this, you just replace the fake data with the actual data to check how it works in the real system. So, for the above problem, we can use a glass of water to try out our programme.

### To start building our program, we need a few things.

1. A microcontroller — One of the most commonly used microcontrollers is Arduino. There are various types of Arduinos available in the market. A commonly used Arduino is called Arduino Uno. Uno can solve our current problem. But remember, we also thought about extending the idea. So now we know that at some point of time, we would need some set of data exchanges between a microcontroller and a mobile application. It's not impossible to do that with Uno, but you need to purchase separate modules for Bluetooth or WiFi connections. (WiFi is needed because at some point we will be controlling the system from anywhere in the world). So keeping the extended idea in mind, we will start with a different type of microcontroller called ESP32. This one comes with a WiFi module installed and costs the same as an Arduino Uno

2. A sensor — We can use a moisture sensor but they tend to be on the expensive side. So for this problem we will try the approach of using sound reflection. For that we will use a HC-SR04 Ultrasonic Sensor.

3. Some jumper wires — Get a few male to female and female to female jumper wires.

We are going to code three things after we connect the sensor with the microcontroller and hopefully that will get our prototype ready. We will do this step by step and create it together. Since the objective is to make you solve the problem, this article will help you with some codes and ideas but at the end you have to do it on your own. These are the basic steps you can follow:-

**Step 1:** Code to check if basic feedback is coming from the sensor.

**Step 2:** Calculate the distance traveled by a pulse emitted by the sensor by measuring the duration between emitting and receiving.

**Step 3:** Put appropriate conditions to trigger some actions based on the distance.

Before that, connect the HC-SR04 Ultrasonic Sensor and the ESP32 as shown in the figure below.

The code shown below will help you with Step 1 and Step 2. You can use the freely available and simple Arduino IDE to write the program.

Let us quickly understand what is happening. First we are assigning the trigger (send) and the echo (receive) pins. You can check the GPIO pin allocation for an ESP32 and assign any pin. For this example, I have chosen GPIO 5 and 18. Then there is the SOUND\_SPEED variable which saves the velocity of sound in the air at 20°C in cm/uS.

I will skip the next few lines as they are self explanatory — we are just initialising some variables where we will be storing our desired values. In the setup() method, we are assigning serial communication at a baud rate of 115200 so that we can print the values on the Serial Monitor. We are defining triggerPin as the output because this actually sends an ultrasound and echoPin as the input as it will receive the reflected ultrasound. Based on this reflection and other parameters we will be calculating the distance later. (because as the water level increases in the tank, the distance will gradually decrease)

In the loop(), we are producing a 10uS HIGH pulse on the triggerPin which helps it to emit an ultrasound.

Next we use the pulseIn() method to calculate the time taken by the ultrasound to be received by the echoPin.

The rest of the part is very simple. We calculate the distance and then print on the serial monitor. Below is the continuous output.

Now comes the actual experiment to see if our model works. At this point, the ultrasound sensor is actually attached at the top of the glass as shown in the diagram.

If you start pouring water in the glass, you should see the distance reducing.

That's it. All that is remaining is step 3, wherein you need to decide the optimal height of the water level, that is the height at which you should receive or hear a notification to stop the pump. This we will leave up to you to figure out but overall this is how the code is going to look like.

You would need to buy an Active Buzzer Module which will cost you around Rs 30 and add to the ESP32 for receiving the sound feedback. Instead of buzzAlarm() you can create a function sendSMS() which will send an SMS to the phone when the desired height is reached

You would need an internet connection for this and would have to also include the WiFi module of ESP32 to connect to the WiFi using SSID and password. The code below (or something along similar lines) will help you with connecting to the WiFi. But after that you need to use some external libraries like TinyGSM to send SMS.

If you have got this running, congratulations! You have just solved one of the most common problems in our “not so developed” cities. But also, the real fun begins. Remember the

possibilities you looked into before? Where you did not have to switch off the pump yourself? Even that can be automated. So now instead of sendSMS(), you need to write a switchOffPump() method. For this you might need some Relay switches added to the Main power via the ESP32 .

As the last step, share your work. Enter the Open source world. Just create a repository on github and let others also look into your work. The problem that you have just solved may be a pain point for someone else as well. That person can help you fine tune your work or could even provide a different point of view. Someone could even build on top of what you have built to solve larger problems. This is the satisfaction you get of contributing to the Open Source world.

No problem is a small problem. It is just a change of perspective that makes a problem small or big. Therefore, do not hesitate to get into this world of IoT (Internet of Things) to solve your problems. This will not only help you achieve something probably no one has achieved before but the solution could also be scaled up to solve bigger issues. To give you an example, our water level checker, at a glance might seem like solving a very customised individual problem, but think about it — Is the notification of the height of water only important for a water tank? Consider the Assam floods, particularly the city of Silchar in Assam. The citizens of this city are suffering with no food or proper water with the hope that the water level of the river Barak, in the valley of which the city is situated, lowers as soon as possible. But can you believe in the year 2022, when technology has become so advanced, people of that city still need to go to a specific place, overcoming the chest height water, just to read the daily reading of the water level of the river?

A little extension of our water level problem can help send the daily reading to people on an hourly basis via SMS. So, please keep innovating solutions which at some level or the other could eventually solve bigger problems of our country.

*Jaydeep is the Head of Communities at Thoughtworks, India. He can be reached at jaydeepc@thoughtworks.com via email, @jchakrabarty on twitter and linkedin.*

### Our code of editorial values

**END**

Downloaded from **crackIAS.com**

© Zuccess App by crackIAS.com

