STEMSEL Intermediate Project 10: Hands Free Sensor Tap

Problem

When we turn on a tap, we may be putting germs onto it which can then jump back onto our hands when we turn the tap off. How can we stop this from happening?

Background

On everything around us there are microscopic organisms living their little lives. For the most part they don't affect us, however some of them can make us really sick or even kill us. Most food poisoning is due to these microbes, most typically a type called Salmonella. For this reason it is important that we try to limit our exposure to these dangerous germs as much as possible. What are some of the ways we can do this?

One simple way is to wash our hands before eating, after going to the toilet or after doing something dirty like digging in the garden. This means that we don't introduce germs that we picked up during the day into our bodies via our food, and also means we don't spread them to other people. This is also why it's important to cover your mouth when you sneeze or cough, since the virus that causes your cold can travel through the air and make other people sick.

However, there are certain instances where even more care has to be taken to eliminate germs. I'm sure you don't want the chef in a restaurant kitchen to have dirty hands when he's making your food, and especially in hospitals we don't want any nasty germs at all, since you go to hospital to get better, not to get sicker! People working in these areas would wash not only their hands but their tools and work surfaces to keep them all in a hygienic condition.

However, what would happen if someone with dirty hands turned on a tap? Some of the germs would go from their hand onto the tap, then when they turned the tap off again the germs can go back onto their hands. What we need to make is a hands-free tap to prevent this situation.

You may have seen hands free hygiene devices before, not only taps but soap dispensers and hand dryers as well. Have you ever wondered how they work?

Ideas

If we can't touch the tap, how else can we turn it on? What have we got in our kits that can simulate the different parts of the tap? What if the chef or doctor is in a hurry and forgets to turn the tap off again? Can we use an automatic off feature? How long should the tap stay on, should we be able to adjust it?

Plan

First, we need some way of turning the tap on and off from a distance. Commercial hands-free taps usually use infrared sensors, but we haven't got those in our kits. Instead, we can use the LDRs since they work in a similar way but use visible light instead of infrared, and we can still use them without touching them. When the LDR senses a shadow that someone had waved their hand near the sensor, the tap should turn on if it is off, or if the tap is on then the tap should turn off.

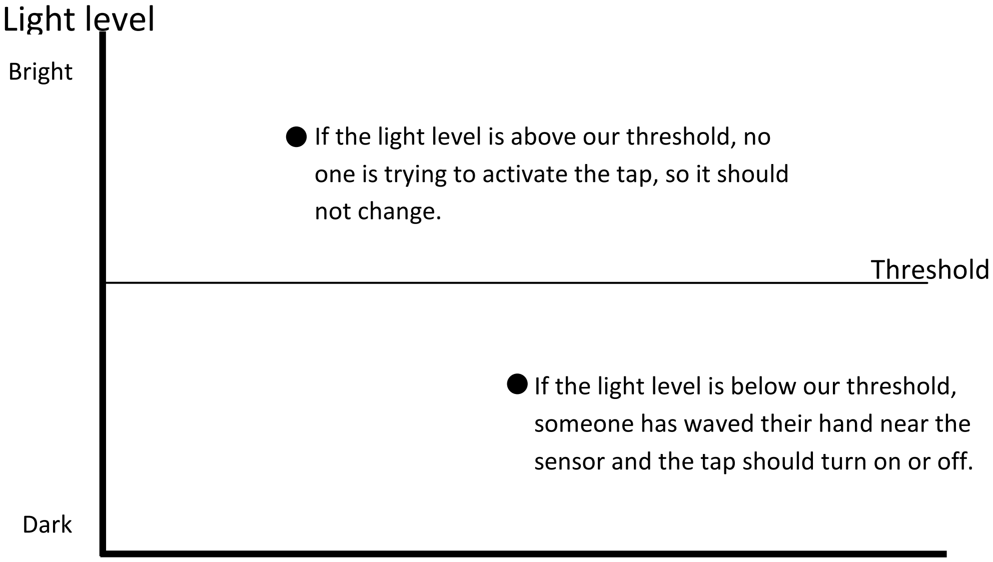

Figure 1: Threshold graph

We should also have something to simulate the tap. The DC motor can represent a pump that would make the water come out of the tap. We will also have an automatic off feature on our tap to help save water. We will use a potentiometer to set how long we want the tap to stay on. Since this project is a bit more complicated than other ones, we will be building it up in several steps.

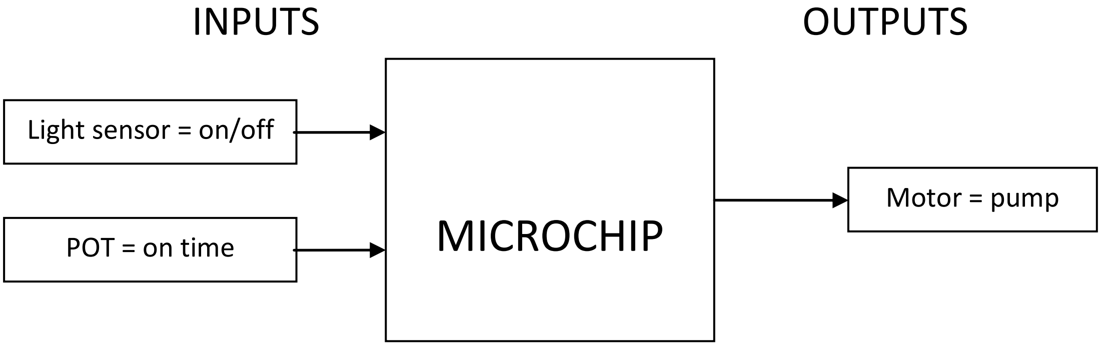

Figure 2: circuit plan

Design

From the plan, we can see that we need to add an LDR, potentiometer and a driver for the pump to the design. Remember to right click to rename the components.

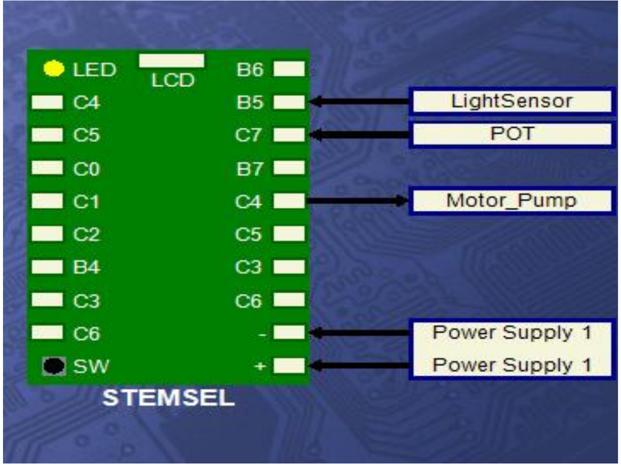

Figure 3: completed circuit design

Build the circuit

Now we can build the circuit. Remember black wires go in negative, red wires go in positive and the white wires go in the pin designated in the design. For the motor, one wire will go into C4, and the other should go into the negative pin.

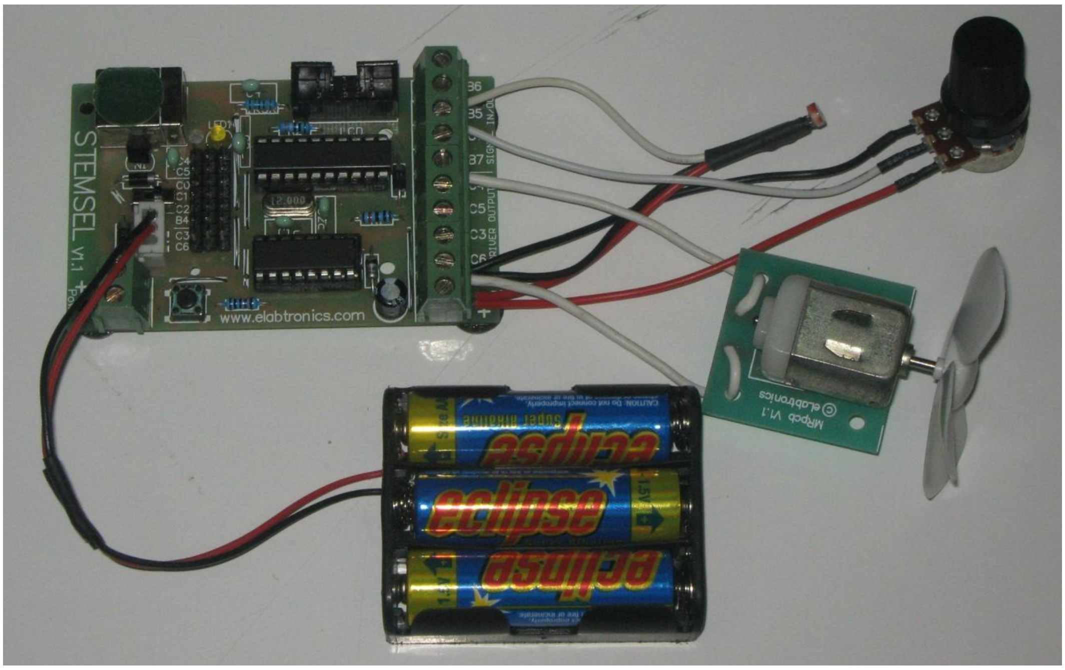

Figure 4: circuit

Programming

Once you have assembled the circuit, send the design to Corechart by clicking the "Send to Corechart" button, then click the "Send Program To Chip" button to program the chip with the LED test program. Once you have verified that the motor is working correctly, delete the test routine so we can start writing our program. Like we said in the plan, we will build the program up in several steps to make sure that each part works ok.

Part A – turning the tap on

1. First, we will get the tap to turn on when it detects a hand. Use an AnalogIn icon to check the light level, then compare it with a threshold value. If the level is above the threshold, there is no hand detected and we want the program to go back to the start.

2. If there is a hand detected, turn the pump on for 10 seconds, then go back to start. Send the program to chip and test it works.

Note that we did not tick the Below box on our compare icon. In this program, we only do something if the light level is below the threshold; otherwise the program just needs to wait. This technique will also be useful later when we need more GOTO icons.

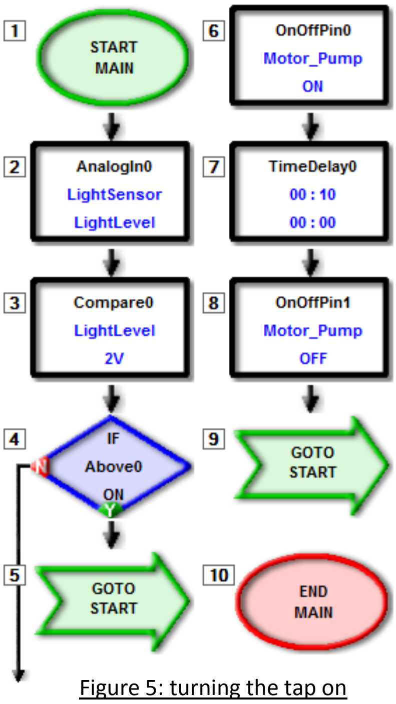

Figure 5: turning the tap on Part A

Part B – timing

3. Next we will use the POT to control the time the tap is on. At the start of the program, use another AnalogInicon to read the value of the POT, saving it asTap_Timer.

4. Now use a Divide icon (Numbers -> Divide) to divide Tap_Timer by 10. This will give a value from 0-25 which we will use for our time delay (remember the POT gives a number between 0 and 255).

5. Modify the time delay so it is Tap_Timer seconds long instead of 10. Send the program to chip and check that you can use the POT to adjust the time the tap is on for.

Part C – turning the tap off

6. Next we will just try using our hands to turn the tap off as well, but there will be no auto-off function. Modify the time delay to 1 second.

7. We will want the program to constantly check if someone has waved their hand near the sensor, so it will be necessary to have two checking loops. There is already one set up to turn it on, but we will also need one to turn it off again. Add an address calledCheckLoop below the time delay.

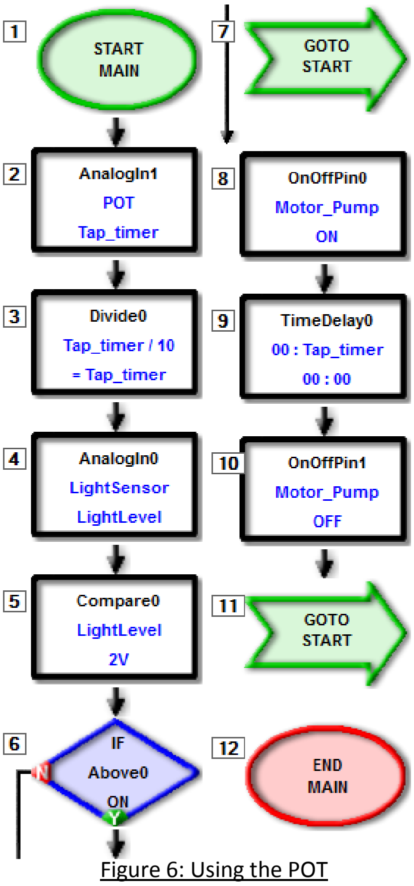

Figure 6: Using the POT

Part B

8. We will need to check the light level again when the tap is on. After the CheckLoop address, read the light level again and compare it with the threshold value. If it is above, we want to check again. Otherwise the program can continue and turn the tap off and go back to start.

9. Put in another 1 second time delay after turning the pump off. Send the newest version to the chip and check that you can turn the tap on and off.

Question: What would happen if the time delays weren't there? How many times a second would the chip check the light level?

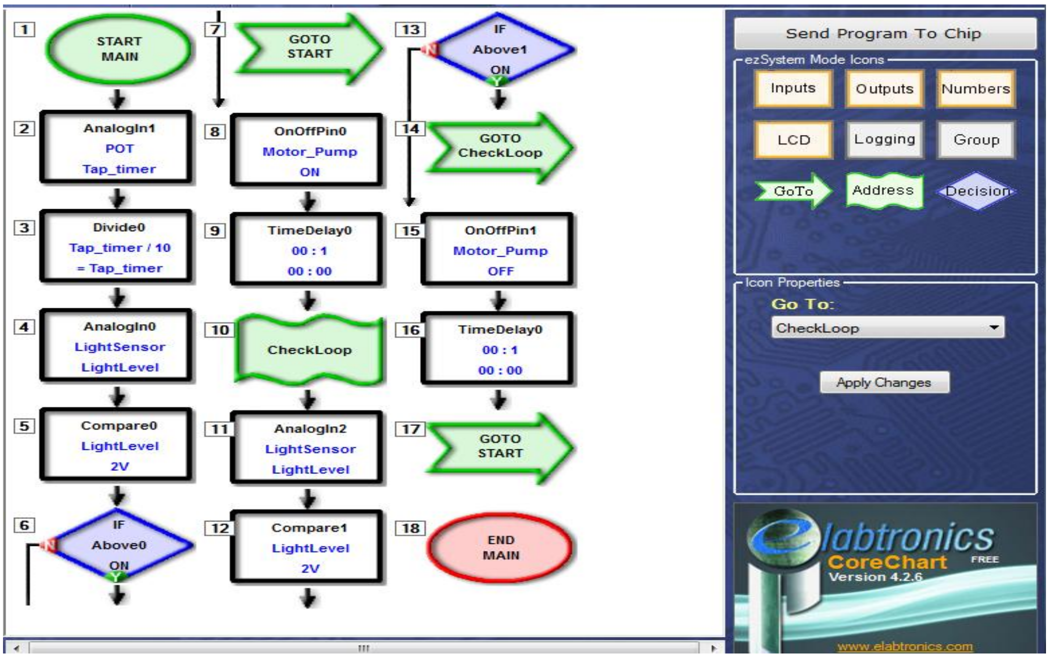

Figure 7: turning the tap on and off - Part C

Part D – full tap

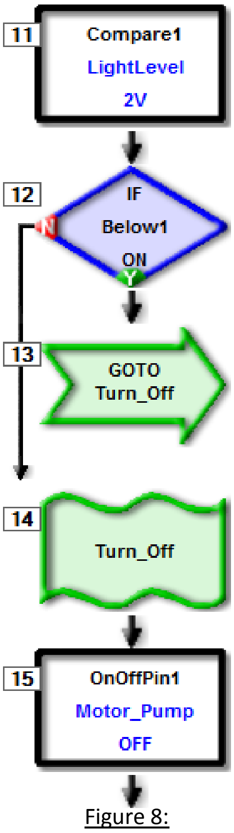

Figure 8: Step 10 and 11

10. Finally, it is time to put all of these parts together so that we can turn our tap on and off, but it also has an adjustable auto-off function. First, delete the Divide icon, since we won't need it. Next, add an address just above the Motor_Pump Off icon called Turn_Off.

11. Change the second Compare icon and the associated GoTo icon so that it still compares LightLevel with the threshold, but instead will go to Turn_Off if the light level is below the threshold.

12. How can we implement the auto-off function? If we use a long time delay, the chip can't check the light level since it is busy counting. Instead, we will need to use a short time delay, and repeat it until the desired time is reached. We will use a delay of ten hundredths of a second, or 0.1 seconds. So if we wanted a delay of 10 seconds like in the first part, we would need to loop through 100 times (0.1 x 100 = 10).

13. Just above the Turn_Off address, add a time delay of 10 hundredths of a second. Then, use a Subtract icon (Numbers -> Subtract) to subtract 1 from Tap_Timer, saving the result as Tap_Timer again.

14. Below this, add another Compare icon, and use it to compare Tap_Timer with 0 (there is no need to put volts, we just want to compare with the number 0). Tick the Above checkbox, and if it is above go back to CheckLoop. The program is complete, so send it to chip.

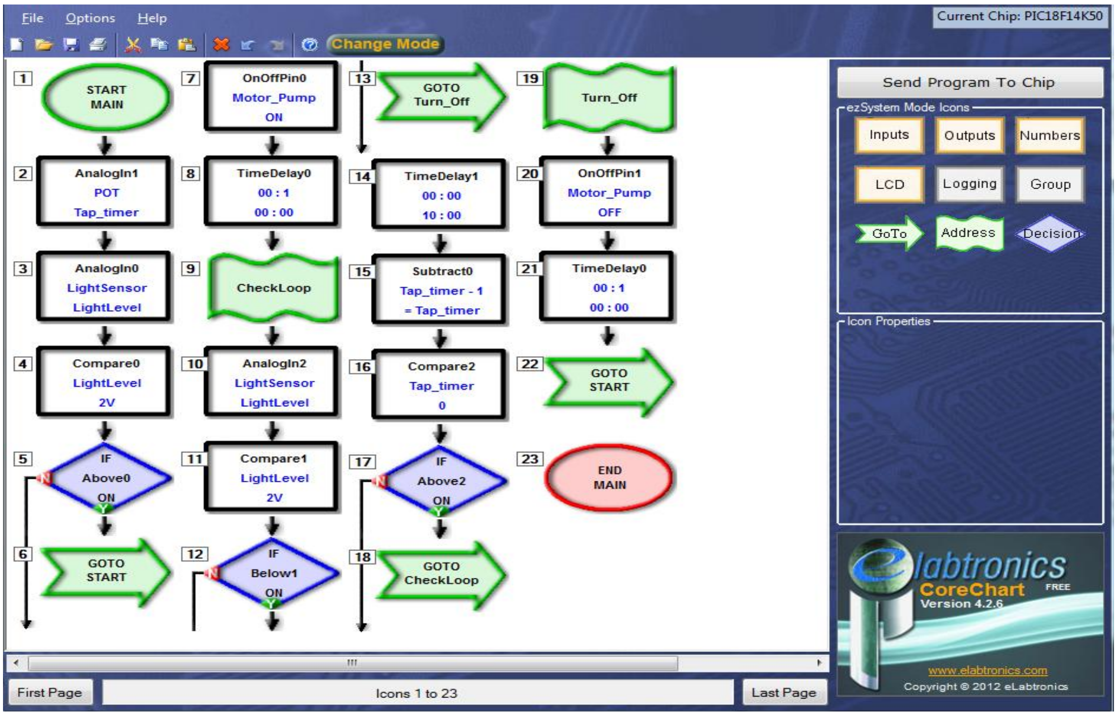

Figure 9: completed tap program

Now the program will just loop at the start until someone waves their hand over the sensor to make it dark. Once this happens, the pump is turned on for 1 second and then the program checks to see if we want the tap to turn off again. At this point, it can do one of two things:

- If someone has waved their hand again, the program will jump to the Turn_Off address to turn the pump off, wait a second and go back to START to check if we want to turn the pump on again.

- If it is still bright, then the program will delay for ten hundredths of a second and subtract one from Tap_Timer. If Tap_Timer is above 0 the program will loop back to CheckLoop to check the light level again. However, if Tap_Timer reaches zero it means the time limit is up and the program will automatically turn off the tap, wait a second and then go back to START.

Remember that Tap_Timer can take any value between 0 and 255, so let's say it is currently set at 80 and no one is there to turn the tap off again. When the program reaches the Subtract icon, Tap_Timer will become 79, which is above 0 so the program will loop back. On the next loop Tap_Timer will become 78, then 77, and so on until it finally reaches 0. In order to reach 0 the program would have looped through CheckLoop 80 times, meaning the total time that the tap was on was the initial delay of 1 second plus 80x0.1 = 8 seconds, i.e. the tap was on for 9 seconds. This means we can adjust the tap to stay on for any time between 1 second and 26.5 seconds.

Bonus Question

Our tap uses a light sensor to sense the shadow of someone's hand. This would be fine while the room is brightly lit, but what would happen when everyone went home and turned the lights off? You can simulate this by leaving your finger over the sensor. This is clearly not desirable, since it would waste a lot of water. How can we modify the program to improve it?

Hint: when we wave our hand over the sensor, it would only be dark for a short time, but if the lights are turned off it would be dark for a long time.

We could modify our program to check if it has been dark for a long time, say 5 seconds. If so, then the microchip should wait until it becomes light again before it starts checking to turn the tap on again.

Summary

It is a fact of life that we are surrounded by micro-organisms, and while most are harmless, there are some that can make us very sick. We can protect ourselves from germs by practicing good hygiene, and one simple way to do this is by washing our hands. However, in certain situations it is vitally important that there are no germs at all, so a hands-free system is required. It is also important to try to save water where possible, so in this project we made a hands free tap that also has an adjustable auto-off feature so we can fight germs and save water at the same time!