Automatic Water Pump Code Transcript

First, we initialize the LCD. This is done by creating an object ‘lcd’ and then adding the I2C address of the display followed by the sample space for display. Which in this case is 16X2.

We declare the variables before the function.

1. duration: datatype: long

function: stores the time in which the HCSRO4 receives the signal back.

1. inches: datatype: long

function: stores the distance between the water level and the HCSRO4.

1. setval: datatype: integer

function: sets the base value of water.

1. percentage: datatype: integer

function: calculates the percentage of water

1. state: datatype: Boolean

function: It stores the value of the manual switch which can be flipped to shift the mode of operation from auto to normal and vice versa

1. pump: datatype: Boolean

function: stores the state of the pump, i.e., ON or OFF

Now we define a function, setup() of return type void. In this function we setup the LCD display.

Lcd.init() : initiates the LCD to receive the signals from the Arduino board.

Lcd.backlight(): Turns on the backlight of the LCD

Lcd.print(“WATER LEVEL”): Prints the statement “WATER LEVEL” on the first line of the LCD.

Lcd.setCursor(0,1): Sets the cursor of the LCD to the next line to print the next statement.

Lcd.print(“PUMP:OFF MANUAL”): Prints the statement within the brackets. This is the initial state of the pump, where the pump is in manual mode and it is currently switched off.

In the next segment, we define the pins of the Arduino board.

Pin 2 is set for output

Pin 3 is set for input

Pin 10 is set for Input Pull Up, which means that it takes the signal from the attached component and that is transferred to another component which will react on the basis of the signal, i.e. receive a HIGH signal or a LOW signal

Pin 11 is also set for Input pull up.

Pin 13 is set for providing OUTPUT

In the next line, the HCSRO4 records the initial reading and sets it to a minimum value.

Now we define a function loop() of return type void.

In this function we first trigger the HCSRO4. We send a HIGH signal to the rangefinder and provide a delay of 10 microseconds, i.e., the HCSRO4 is triggered every 10 microseconds.

Now we record the data received by the HCSRO4. The time is stored in the variable duration.

A function, microsecondstoInches() which is defined later is used to convert the time recorded in the duration variable to inches which helps in storing the distance between the water level and the HCSRO4

Next the percentage of water in the container is printed. This value will be updated in the course of the program.

Now the percentage of water filled is calculated. This is done by deducting the inches recorded from the initial height of the container we have, and then converting that distance to a percentage value. This gives us how much water has been filled.

The percentage of water is then checked through a conditional statement. If the level of water is below 30%, then the state of the pump variable is set to 1, i.e., the pump has to be turned ON. If the percentage of water in the container id more then 85%, the pump variable is set to 0, i.e., switching the pump OFF.

The cursor is set to the next line in the LCD. A conditional statement checks the state of the pump variable and prints likewise, whether if it is ON or OFF.

Now the input from the Pin 11 is read. The Pin 11 is connected to the external physical switch, which helps to switch between the AUTO mode and MANUAL mode of the pump.

The input from the Pin 10 and 11 is again read as a part of another conditional statement. If the reading is 0, then the state is set to 1 and the value of the pump variable is reversed.

So if the switch is in MANUAL mode, then it does not send any auto signal to the pump, the pump continue to flow water even when the level crosses 85%, as long as the Push button is held down.

If it is in AUTO mode, no external interference is required, the pump will continue pumping water till the level of the water reaches 85% of the capacity of the container.

In the last part, we define a function, microsecondstoinches() which we have called above. This function converts the time duration taken for the signal to be transmitted form the HCSRO4 till being received by the sensor from microseconds to inches. Then the value is halved, as the time is recorded for the round trip of the signal, and we need to calculate the depth.

End of Code