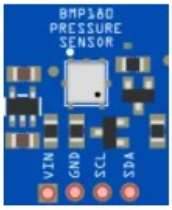
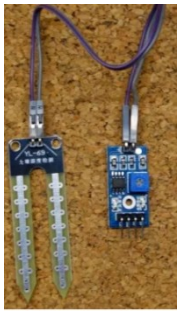
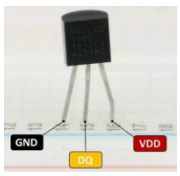
INTERNET OF THINGS
DIGITAL ASSIGNMENT – 1

Name:- Mohit Kumar
Reg.No.- 17BCE0067

1.
List the different types of sensors available (not less than 15) and create a table mentioning sensor image, its usage, connection specifications, etc. An example information for one sensor is given in the following table.

SI.NO	IMAGE	SENSOR NAME AND IC	USAGE	CONNECTION SPECIFICATION	OTHER INFORMATION
1.		DHT11/ DHT22 Humidity and Temperature Sensor	The DHT11 sensor is used to measure temperature and relative humidity	4 pins in DHT sensors. 1. 5V 2.Any digital GPIO; also connect a 10k Ohm pull-up resistor 3.Don't connect 4.GND	These sensors contain a chip that does analog to digital conversion and spit out a digital signal with the temperature and humidity.
2.		BMP180 Barometric Sensor	It measures the absolute pressure of the air around it.	4 pins Vin 5V GND GND SCL A5 SDA A4	It can also measure altitude and temperature. The BMP180 barometric sensor communicates via I2C interface.

3.	 Rain Sensor	Rain Sensor FC-37	The rain sensor is used to detect water and it can detect beyond of what a humidity sensor do	4 pins 1,A0 Analog pins 2.D0 Digital pins 3.GNDGND 4.VCC 5V	The rain sensor has a built-in potentiometer for sensitivity adjustment of the digital output (D0). It also has a power LED that lights up when the sensor is turned on and a digital output LED.
4.		Soil Moisture Sensor YL-69	The soil moisture sensor or the hygrometer is usually used to detect the humidity of the soil. So, it is perfect to build an automatic watering system or to monitor the soil moisture of your plants.	4 pins 1.A0 Analog Pins 2.D0 Digital Pins 3.GNDGND 4.VCC 5V	The voltage that the sensor outputs changes accordingly to the water content in the soil. When the soil is: Wet: the output voltage decreases Dry: the output voltage increases
5.		DS18B20 Temperature Sensor	The DS18B20 temperature sensor is a one-wire	3 pins 1.GNDGND 2.DQ Any digital pin	Each DS18B20 temperature sensor has a unique 64-bit

			digital temperature sensor	(with 4.7k Ohm pull-up resistor) 3.VDD5V (normal mode) or GND (parasite mode)	serial code. This allows us to wire multiple sensors to the same data wire. So, we can get temperature from multiple sensors using just one Arduino digital pin.
6.		Real Time Clock (RTC) DS1307	The module has a backup battery installed. This allows the module to retain the time, even when it's not being powered up by the Arduino.	4 pins 1.SCL A5 2.SDA A4 3.VCC 5V 4.GNDGND	This module uses I2C communicati on. This means that it communicate s with the Arduino using just 2 pins.
7.		MQ-2 Gas/Smoke Sensor	The MQ-2 smoke sensor is sensitive to smoke and to the following flammable gases: LPG Butane Propane Methane Alcohol Hydrogen	The MQ-2 sensor has 4 pins. 1.A0 Analog pins 2.D0 Digital pins 3.GNDGND 4.VCC 5V	The voltage that the sensor outputs changes accordingly to the smoke/ gas level that exists in the atmosphere. The sensor outputs a voltage that is proportional to the concentration of

					smoke/gas.
8.		HC-SR04 ultrasonic sensor	The ultrasonic sensor uses sonar to determine the distance to an object.	VCC: +5VDC Trig : Trigger (INPUT) Echo: Echo (OUTPUT) GND: GND	It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package. It comes complete with ultrasonic transmitter and receiver modules.
9.		PIR Motion Sensor	The PIR motion sensor is ideal to detect movement. PIR stand for “Passive Infrared”. Basically, the PIR motion sensor measures infrared light from objects in its field of view.	the sensor has only 3 pins. GND – connect to ground OUT – connect to an Arduino digital pin 5V – connect to 5V	It can detect motion based on changes in infrared light in the environment. It is ideal to detect if a human has moved in or out of the sensor range.
10.		Tilt Sensor	The tilt sensor allows to detect orientation or inclination. It detects if the sensor is completely upright or if	One pin to an Arduino digital pin and GND to GND.	The tilt sensor acts like a switch that is turned on or off depending on its inclination. So, it will

			it is tilted.		give digital information to the Arduino, either an HIGH or a LOW signal.
11.		MFRC522 RFID Reader	RFID means radio-frequency identification . RFID uses electromagnetic fields to transfer data over short distances. RFID is useful to identify people, to make transactions	1.SDA Digital 10 2.SCK Digital 13 3.MOSI Digital 11 4.MISO Digital 12 5.IRQ unconnected 6.GNDGND 7.RST Digital 9 8.3.3V 3.3V	You can use an RFID system to open a door. For example, only the person with the right information on his card is allowed to enter
12.		SRD-05VDC-SL-C 5V Relay	A relay is an electrically operated switch that can be turned on or off, letting the current go through or not, and can be controlled with low voltages, like the 5V	The set at the right consists of VCC and GND to power up the module, and input 1 (IN1) and input 2 (IN2) to control the bottom and top relays, respectively.	This relay module has two channels (those blue cubes). There are other models with one, four and eight channels. This module should be powered with 5V,

			provided by the Arduino pins.	The second set of pins consists of GND, VCC, and JD-VCC pins. The JD-VCC pin powers the electromagnet of the relay.	which is appropriate to use with an Arduino.
13.		nRF24L01 – 2.4GHz RF Transceiver Module	The nRF24L01 is used on a wide variety of applications that require wireless control. They are transceivers which this means that each module can transmit and receive data.	1.GND 2.VCC 3.CE 4.CSN 5.SCK 6.MOSI 7.MISO 8.IRQ	The module can use 125 different channels which gives a possibility to have a network of 125 independently working modems in one place. Each channel can have up to 6 addresses, or each unit can communicate with up to 6 other units at the same time.
14.		ACS712 Current Sensor	Sensor module for measuring current. Both AC and DC, up to 5A can be measured. The output is a linear voltage	1 & 2 IP+ +ve terminals for sensing current 3 & 4	Sensing terminal can even measure current for loads operating at high voltages like 230V AC mains

			between 0 – VCC, centered around VCC/ 2 at 0A. The load is connected in series to the screw terminal.	IP- -ve terminals for sensing current 5 GND Signal Ground 6 FILTER External Capacitor (to set the bandwidth) 7 VIOUT Analog Output 8 VCC Power Supply	while output sensed voltage is isolated from measuring part.
15.		proximity sensor	This is a multipurpose infrared sensor which can be used for obstacle sensing, color detection(between basic contrasting colors), fire detection, line sensing, etc and also as an	IR Sensor have three to four Lines 1. +5V VCC 2. GND 3. D0 or OUT (Digital Output) 4. A0 - Analog Out	The sensor outputs a logic one(+5V) at the digital output when an object is placed in front of the sensor and a logic zero(0V), when there is no object in front of the sensor. An on

			encoder sensor. The sensor provides a digital output.		board LED is used to indicate the presence of an object.
16.		Microphone Sound Sensor	The microphone sound sensor, as the name says, detects sound. It gives a measurement of how loud a sound is.	A0 Analog pins D0 Digital pins GND GND VCC 5V	The compactor ic sensor modules have a built-in potentiometer to adjust the sensitivity of the digital output pins. High-sensitivity sound detection module with 2 outputs. AO = Analog output, real-time output voltage signal of the microphone. DO = The digital output depends on the sound intensity and the threshold that has been set.
17.		Color Sensor-TCS230	he TCS230 sensor senses color light with the help of an 8 x 8 array of photodiodes.	GND=4 Power ground OE=3 Enable for	The full-scale output frequency can be scaled by one of three preset values via

			Then using a Current-to-Frequency Converter the readings from the photodiodes are converted into a square wave with a frequency directly proportional to the light intensity.	Output Frequency (Active Low) OUT=6 Output Frequency S0,S1=1,2 Output Frequency scaling selection inputs S2,S3 = 7,8 Photo diode type selection inputs VCC-5 Voltage supply	two control input pins. Digital inputs and digital output allow direct interface to a micro-controller or other logic circuitry. Output enable (OE) places the output in the high-impedance state for multiple-unit sharing of a micro-controller input line.

2.

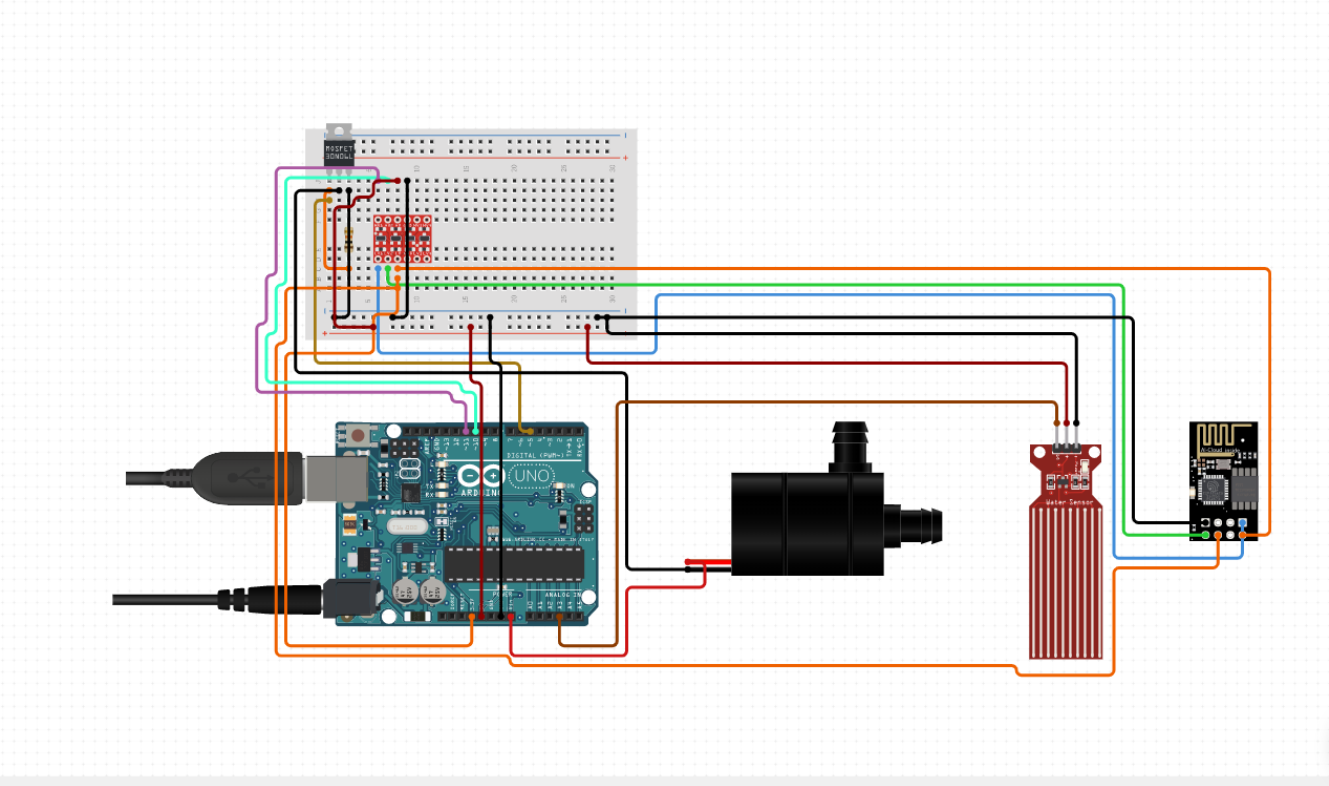
Design a module for smart city, named smart municipal water release approach. The tank should have two sensors one is to identify the maximum tank capacity and the other is to maintain the minimum water level as a backup. The tank should automatically release the water at a particular time on every day. The water which is released by tank should be noted in online cloud. Consider ThinkSpeak / Amazon Web Service (AWS) for this process. Design the same using any simulator (ThinkCad) / NodeRed / others and display the diagram here.

Inventory:-

1. Arduino uno
2. Bread Board – Half Size
3.Submersible Pool Water Pump
4. Water Flow Sensor G1/2

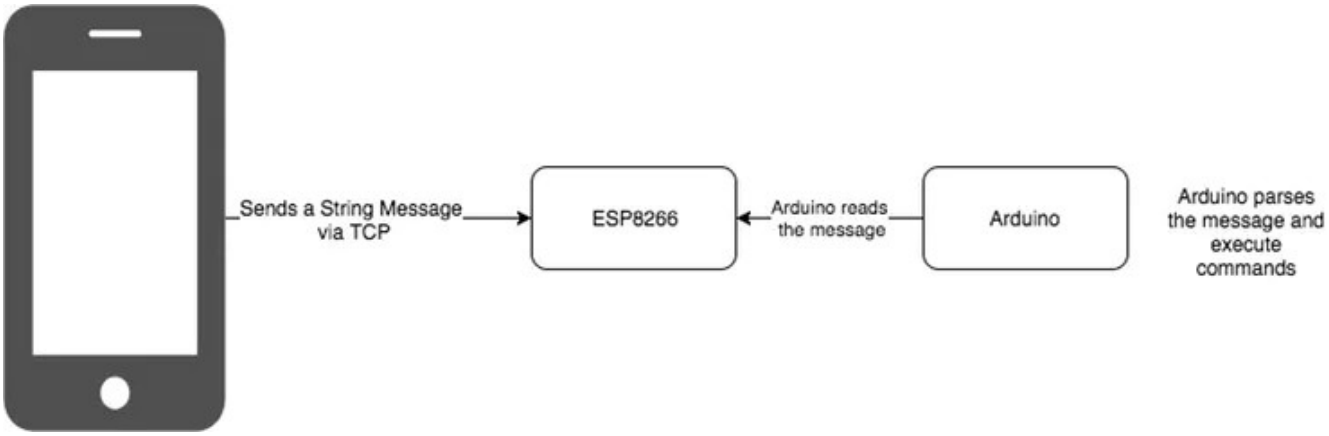
5. Water lever Sensor Moduel
6. Wifi Moduel (ESP-8266)
7. Jumper Wires
8.Logic Level Converter Bi-Directional
9. 10K Ohm Resistor
10.N-Channel MOSFET 60V 30A

Circuit Diagram:-



Reference:- The Diagram was created using the website “circuit.io”.

Communication Process for ESP8266



Cloud Connectivity:- We store the data read by the sensors on the cloud using things speak.

Code:-

```
 #include <SoftwareSerial.h>

SoftwareSerial espSerial =  SoftwareSerial(2,3);
 // arduino RX pin=2  arduino TX pin=3    connect the arduino RX pin to esp8266 module TX pin   - connect the arduino TX pin to esp8266 module RX pin

float Water_flow =A0; //water level sensor
float Water_pressure= A1; //Water Pressure Sensor
float level_value;
float flow_value;
int motor =5;

String apiKey = "5G0NEIEWV7FQWERTY"; // replace with your channel's thingspeak WRITE API key
String ssid="kavin";    // Wifi network SSID
String password ="12345612";  // Wifi network password
boolean DEBUG=true;
#define VOLTAGE_MAX 5.0
#define VOLTAGE_MAXCOUNTS 1023.0

//======================== showResponse====================================

void showResponse(int waitTime){
long t=millis();
char c;
while (t+waitTime>millis()){
if (espSerial.available()){
```

```
c=espSerial.read();
if (DEBUG) Serial.print(c);
}
}
}
//======================connection to
thinkspeak.com================================================
boolean thingSpeakWrite(float value1){
String cmd = "AT+CIPSTART=\"TCP\",\"";                  // TCP connection
cmd += "184.106.153.149";                               // api.thingspeak.com
cmd += "\",80";


espSerial.println(cmd);

if (DEBUG) Serial.println(cmd);

if(espSerial.find("Error"))
{
if (DEBUG) Serial.println("AT+CIPSTART error");
return false;
}
String getStr = "GET /update?api_key=";   // prepare GET string
getStr += apiKey;
getStr +="&field1=";
getStr += String(value1);
//getStr +="&field2=";
//getStr += String(value2);
// ...
getStr += "\r\n";
// send data length
cmd = "AT+CIPSEND=";
cmd += String(getStr.length());
espSerial.println(cmd);

if (DEBUG)  Serial.println(cmd);
delay(100);
if(espSerial.find(">"))
{
espSerial.print(getStr);
if (DEBUG)  Serial.print(getStr);
}
else
{
espSerial.println("AT+CIPCLOSE");
// alert user
if (DEBUG)
{  Serial.println("AT+CIPCLOSE");
return false;
```

```
}
return true;
}

//==============================setup====================================

void setup() {

DEBUG=true;          // enable debug serial

//----------my sensor code-----------

pinMode(LED_BUILTIN, OUTPUT);
pinMode(motor, OUTPUT);
pinMode(Water_flow,INPUT);
pinMode(Water_pressure,INPUT)
Serial.begin(9600);
espSerial.begin(115200);  // enable software serial



// Your esp8266 module's speed is probably at 115200.
// For this reason the first time set the speed to 115200 or to your esp8266 configured speed
// and upload. Then change to 9600 and upload again

//espSerial.println("AT+RST");         // Enable this line to reset the module;
//showResponse(1000);

//espSerial.println("AT+UART_CUR=9600,8,1,0,0");    // Enable this line to set esp8266 serial speed to
9600 bps
//showResponse(1000);
espSerial.println("AT+CWMODE=1");   // set esp8266 as client
showResponse(1000);

espSerial.println("AT+CWJAP=\""+ssid+"\",\""+password+"\"");  // set your home router SSID and
password
showResponse(5000);

if (DEBUG)  Serial.println("Setup completed");
}
// ==============================loop==================================
void loop() {
//---------my gas sensor value reading-------------------
level_value= analogRead(Water_flow);
flow_value=analogRead(Water_pressure);
delay(1);  //delay in between reads for stability

float t = level_value(VOLTAGE_MAX / VOLTAGE_MAXCOUNTS);
```

```
float f = flow_value(VOLTAGE_MAX / VOLTAGE_MAXCOUNTS);

thingSpeakWrite(t);             // Write values to thingspeak

// thingspeak needs 15 sec delay between updates, */
delay(20000);
thingSpeakWrite(f);
delay(20000);

//motor condition
if(t<=80){
digitalWrite(motor,HIGH);
}else{
digitalWrite(motor,LOW);
}

}
```