// The ifference betweeen the thinger-update.ino,

//I defined the 24 as output and 22 for input for the ultrasonic part and defined the maximum range

#define \_DEBUG\_

#include <ThingerEthernet.h>

#include<ThingerClient.h>

#include "EmonLib.h"

#include <Ultrasonic.h>

#define USERNAME "husamgh"

#define DEVICE\_ID "ard1"

#define DEVICE\_CREDENTIAL "Wrq$@qJct&tE"

ThingerEthernet thing(USERNAME, DEVICE\_ID, DEVICE\_CREDENTIAL);

// Include Emon Library

EnergyMonitor emon1; // Create an instance

Ultrasonic ultrasonic(24,22);//24 trig,22 echo,

int sum;

float Irms, Vo;

float flowRate;

//---- for flow meter

volatile int flow\_frequency; // Measures flow sensor pulses

unsigned int l\_hour; // Calculated litres/hour

unsigned char flowsensor = 2; // Sensor Input

unsigned long currentTime;

unsigned long cloopTime;

// for the thermal resistor

const float resistance = 10000;

float lecture\_echo;

double distance,Dist,total;

void setup() {

Serial.begin(9600);

emon1.current(13,30);// Current: analog input pin, calibration.

thing["valve"] << [](pson& in) {if(in.is\_empty()){

in = (bool) digitalRead(37);

}

else{

digitalWrite(37, in ? HIGH : LOW);

}};

thing["pump\_activate"] << [](pson& in) {if(in.is\_empty()){

in = (bool) digitalRead(35);

}

else{

digitalWrite(35, in ? HIGH : LOW);

}};

thing["SONIC"] >> [] (pson& out){

Dist= ultrasonic.read();

//Dist=155-distance;

total=1000-(1000\*(Dist/155));

out=total;};

thing["SONIC\_2"] >> [] (pson& out){

distance = ultrasonic.read();

out=distance;};

thing["water pump"] >> [](pson & out) {

Irms = emon1.calcIrms(2000);

out = Irms;};

thing["Flow meter"] >> [](pson & out) {

currentTime = millis();

// Every second, calculate and print litres/hour

if(currentTime >= (cloopTime + 1000))

{

cloopTime = currentTime; // Updates cloopTime

// Pulse frequency (Hz) = 7.5Q, Q is flow rate in L/min.

l\_hour = (flow\_frequency / 7.5); // (Pulse frequency) / 7.5Q = flowrate in L/min

flow\_frequency = 0; // Reset Counter

}

out=l\_hour;

};

pinMode(24,OUTPUT);// Trigger pin

pinMode(22,INPUT);// Echo pin

pinMode(35, OUTPUT);

pinMode(37, OUTPUT);

pinMode(14, INPUT);

pinMode(41, OUTPUT);

digitalWrite(35,HIGH);

digitalWrite(37,HIGH);

digitalWrite(41,HIGH); // to show that it is active

//---flow meter

pinMode(flowsensor, INPUT);

digitalWrite(flowsensor, HIGH); // Optional Internal Pull-Up

Serial.begin(9600);

attachInterrupt(0, flow, RISING); // Setup Interrupt

sei(); // Enable interrupts

currentTime = millis();

cloopTime = currentTime;

}

void loop() {

thing.handle();

}

void flow () // Interrupt function

{

flow\_frequency++;

}