Centralized Heating Water

Official Heat Energy Meter

Heat meter photo is taken by fswebcam once in 15 s. Heat meter blinks, so 5 of 6 images are empty. Click on image to enter archive. Illumination of heat meter:

LEDon **LEDoff**

Heating Energy Saving

Centralized hot-water water valve control using Arduino and DC motor with a reductor. Arduino also controlls 220V power for circulation pump.
By default Arduino opens valve after 15 min, if no close command received from server, so that people will not freeze, if the server crashes or is not present.

Open valve **Close valve**

Arduino

Serial data saved in file. Should change every two seconds.

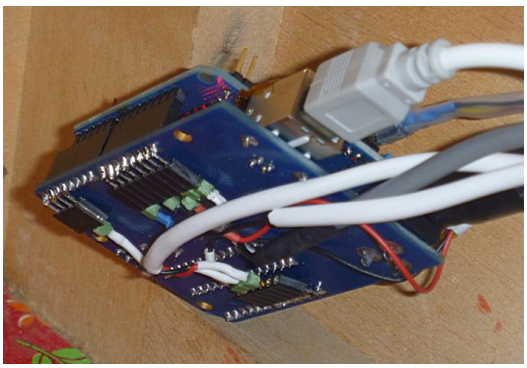
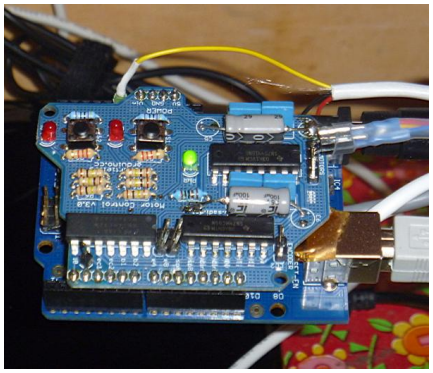
```
databegin 56.31 59.25 54.13 26.44 272 close PUMPoff LEDon 471 dataend
Sv jÅ«n 10 16:25:50 CEST 2012
```

ADC reads potenciometer connected to valve. Allows to check that the motor did what expected. 270 - valve closed. 320 - valve open.


Camera in white plastic pipe. Leds also inside pipe.
White pipe makes scattered diffuse light that is good for meter illumination.

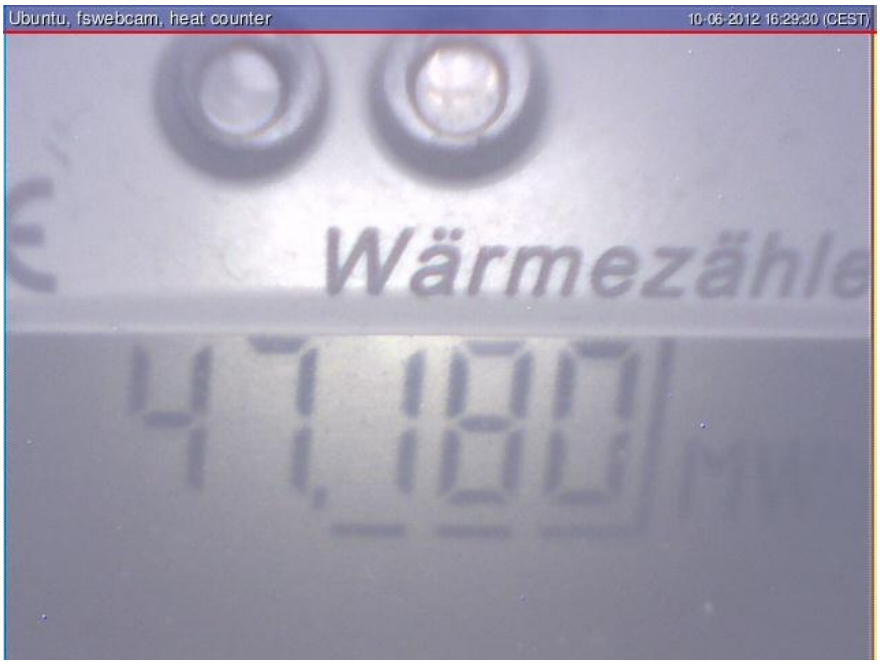

Speed 400 motor with 30 times redactor. From Conrad.de. Potentiometer 100 k with arm to sense how deep is valve.

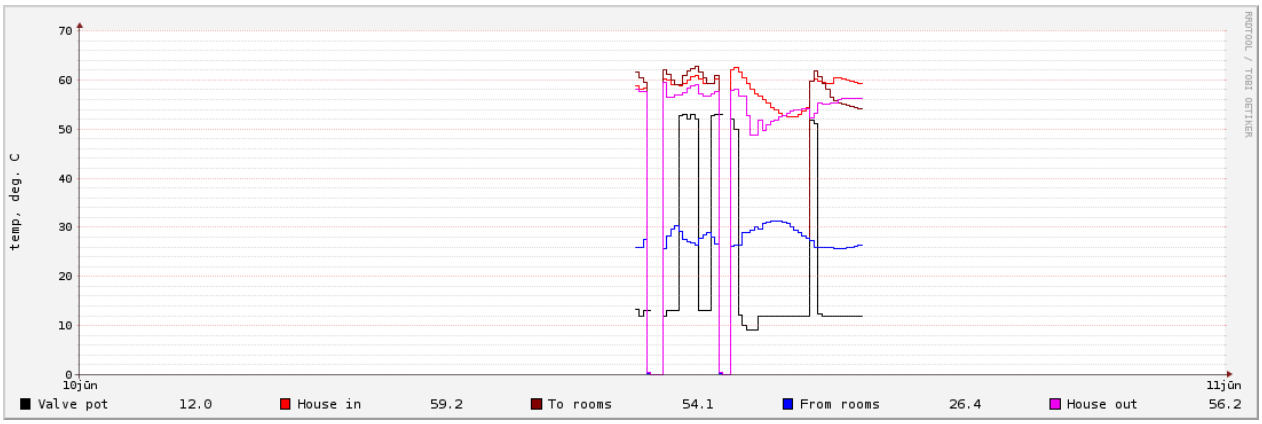
Potentially continuous rotation RC servo can be used instead of motor.

Extra connectors are soldered to Arduino. Motor shield is used.

Can put another webcam to view valve. But is not necessary as potentiometer shows status.


Ubuntu, fswebcam, heat counter
10-06-2012 16:29:30 (CEST)
Wärmezähle
47,180


70
60
50
40
30
20
10
0
temp, deg. C
10jún
11jún
RRDTOOL / TOBI OETIKER
Valve pot 12.0
House in 59.2
To rooms 54.1
From rooms 26.4
House out 56.2

```
#include <OneWire.h>
#include <DallasTemperature.h>
#define TEMPERATURE_PRECISION 12
#define ONE_WIRE_BUS 2 // Onewire data wire is plugged into port 2 on the Arduino. Needs ca 4k pullup resistor to 5V.

OneWire oneWire(ONE_WIRE_BUS); // Setup 1Wire instance to communicate with any OneWire devices (not just Maxim/Dallas temperature ICs)
DallasTemperature sensors(&oneWire); // Pass our oneWire reference to Dallas Temperature.
DeviceAddress Thermometer1, Thermometer2, Thermometer3, Thermometer4; // arrays to hold device addresses

String dig13="LEDoff "; String dig11="PUMPon ";
String tA1="open "; // last command output to motor
int count=0; //byte count in serial message
int i = 0; int dirA = 12; int speedA = 10;

void setup() {
pinMode (dirA, OUTPUT); pinMode (speedA, OUTPUT); //motor PWM
pinMode(11, OUTPUT);  pinMode(13, OUTPUT); // digital outputs that can be switched on with a serial command
pinMode(8, OUTPUT);  digitalWrite(8, 0); //led gnd
 OPEN();
 Serial.begin(9600); Serial.flush();
 Serial.println ("Reads AI0 valve pot, Controls digital output pins 11 - pump and 13 LED. Commands: 11on, 11off, 13on, 13off. Commands to turn valve motor: open, close");
 Serial.println ("Generates 1-wire bus (pin2) for DS18B20 temperature sensors. Needs external 4.7k pullup.");
 Serial.println ("External power 9V for motor needed. Valve opens in the startup and afer 20 min if no command received");
 Serial.println ("Make sure there is no echo from USB host back to Arduino, buffer will overload and commanding will not work");
  sensors.begin();
   if (!sensors.getAddress(Thermometer1, 0)) Serial.println("Unable to find  1-wire sensor nr 1");
   if (!sensors.getAddress(Thermometer2, 1)) Serial.println("Unable to find 1-wire sensor nr 2");
   if (!sensors.getAddress(Thermometer3, 2)) Serial.println("Unable to find 1-wire sensor nr 3");
   if (!sensors.getAddress(Thermometer4, 3)) Serial.println("Unable to find 1-wire sensor nr 4");
 sensors.setResolution(Thermometer1, TEMPERATURE_PRECISION);
sensors.setResolution(Thermometer2, TEMPERATURE_PRECISION);
 sensors.setResolution(Thermometer3, TEMPERATURE_PRECISION);
sensors.setResolution(Thermometer4, TEMPERATURE_PRECISION);
}

void printAddress(DeviceAddress deviceAddress) { for (uint8_t i = 0; i < 8; i++)  {  }  }
```

```
void printTemperature(DeviceAddress deviceAddress) { float tempC = sensors.getTempC(deviceAddress);
Serial.print(tempC); Serial.print(" ");  }
void printResolution(DeviceAddress deviceAddress) {  }
void printData(DeviceAddress deviceAddress) {  printAddress(deviceAddress);
printTemperature(deviceAddress); }

void OPEN(){tA1="open ";  dig11="PUMPon ";  digitalWrite(11, 1); digitalWrite (dirA, HIGH);
   analogWrite (speedA, 255); delay (1000);
   analogWrite (speedA, 0); i=0;}
void CLOSE(){ tA1="close "; dig11="PUMPoff "; digitalWrite(11, 0); digitalWrite (dirA, LOW);
   for (int j = 50; j < 100; j += 10) { analogWrite (speedA, j); delay (300); }
   analogWrite (speedA, 0); i=0;}



void loop() {
String kommand=""; // string received via com port
String report=""; // string output to com port
count=Serial.available();
 for (int x=0; x<count; x++) { char ch = Serial.read(); kommand = kommand + ch;}
//if (kommand != "") {Serial.print (kommand); Serial.println (count);}
     kommand=kommand.substring(0,4);
  if (kommand=="open") {OPEN();  }   if (kommand=="clos") {CLOSE(); }
  if (kommand=="11on") {dig11="PUMPon "; digitalWrite(11, 1);}
  if (kommand=="11of") {dig11="PUMPoff "; digitalWrite(11, 0);}
  if (kommand=="13on") {dig13="LEDon "; digitalWrite(13, 1);}
  if (kommand=="13of") {dig13="LEDoff "; digitalWrite(13, 0);}

Serial.print("databegin ");
sensors.requestTemperatures();  printData(Thermometer1);  printData(Thermometer2);
printData(Thermometer3); printData(Thermometer4);
report = report+analogRead(A0)+" "+tA1+dig11+dig13+i+" dataend";
Serial.println(report);
delay(500);
if ( i > 800 ) {i=0; OPEN();} // 4*300*40/60     20 min
i++;


}
```