```
/*
 <datalogger.ino> Copyright (C) <2015> <Sneha Kopallu>,<Prasad Mehendale>
         This program comes with ABSOLUTELY NO WARRANTY;
         for details, refer to the GPL-3.0.txt on www.gnu.org .
         This is free software, and you are welcome to redistribute
         it under the conditions of GNU-GPL V3.0
*/

/*
 Program : datalogger
 Description :
 1. User will set the time between two successive readings.
 2. Program reads the values set on POTs for time unit and time count
 and will find the time for each reading.
 3. If start key is pressed, it will read the analog input at pin A0.
 4. This digital count is divided by 4 and it will store the value to EEPROM
 delay of the user specified count is given and it will repeat the procedure
untill all the 1023 EEPROM addresses are filled.
 5. After this the controller will turn the LED at pin D13.
 6. To read the data user has to press READ key connected to pin D7.
 7. If READ key is pressed , program will send the values stored in EEPROM to
serial monitor.
 (data collected on serial terminal is in CSV form)

 Hardware details :
 pins used      connected to
 D6             key START
 D7             Key READ
 D13            LED (red)
 A0             Sensor input(through opamp)
 A1             Time count (POT) range 1-9
 A2             Time Unit (POT) range 250us-1hr
 */

#include <EEPROM.h>
#define micro      1
#define milisec    2
#define BIT_8      8
#define BIT_10     10

//
// INPUTs and OUTPUTs
//-------- Digital  ----------------
const int led = 13;
const int READ= 7;     // Read key pin
const int START=6;    //Start key
const int selector = 8;  // key for selector switch
//-------  Analog  ------------------
const int input=A0;  // Analog input pin that the potentiometer is attached to
const int multiplier=A2;  //user can select us to hrs range
const int time_count=A1;  // user can select the time from 1 to 9 unit
//-------- Variables  --------------------------------
int analog_val = 0;        // to read the analog value from analog pin
int map_val;        //to storemap function result
unsigned long time_multiplier;// multiplier selected by user
int time;      //time count selected by the user
unsigned long delay_val;    // to store the delay set by user
int time_unit;
int ee_addr=0;
int pin_state;
int count;
byte sample[1024];
byte storage= BIT_8;
//--------------- Initialization ----------------------------
```

```
void setup () {
  pinMode(led, OUTPUT);  //LED pin output
  pinMode(selector, INPUT);   //selector switch
  pinMode(READ, INPUT); //READ key input
  pinMode(START, INPUT); //START key input
  Serial.begin(9600);  //serial communication enable
  digitalWrite(led, LOW);   //LED OFF
}

void loop () {

  //  ===============Step 1 : Find the delay set by user using two POTs
=================
  //read the analog value of time_count POT
  analog_val = analogRead(time_count);
  //map the digital count to the range 1 to 9
  time = map(analog_val,0,1023,1,10);
  // if it exceeds max count=9
  if(time > 9)
  {
    time = 9;
  }
  //read the analog value for multiplier
  analog_val = analogRead(multiplier);
  map_val=map(analog_val,0,1023,1,10);
  switch(map_val)
  {
  case 1:
    time_multiplier = 250;
    time_unit = micro;
    break;
  case 2:
    time_multiplier = 1000;
    time_unit = micro;
    break;
  case 3:
    time_multiplier = 10;
    time_unit = milisec;
    break;
  case 4:
    time_multiplier = 100;
    time_unit = milisec;
    break;
  case 5:
    time_multiplier = 1000;
    time_unit = milisec;
    break;
  case 6:
    time_multiplier = 10000;
    time_unit = milisec;
    break;
  case 7:
    time_multiplier = 60000;
    //    Serial.println("multiplier");
    //    Serial.println(time_multiplier);
    time_unit = milisec;
    break;
  case 8:
    time_multiplier = 600000;
    //    Serial.println("multiplier");
    //    Serial.println(time_multiplier);
    time_unit = milisec;
    break;
  case 9:
```

```
    time_multiplier = 3600000;
    //    Serial.println("multiplier");
    //    Serial.println(delay_val);
    time_unit = milisec;
    break;
  default :
    time_multiplier = 3600000;
    //    Serial.println("multiplier");
    time_unit = milisec;
    break;
  }
  delay_val = ((long)time)*time_multiplier;
  if(digitalRead(selector) == LOW)
  {
    storage = BIT_10;  //if Selector swich is LOW, 10 bit storage
  }
  else
  {
    storage = BIT_8;  //default storage is 8 bit
  }
  //   Serial.println(delay_val);

  //  ===============Step 1 : Start taking the readings and store it to EEPROM
if START key is pressed============
  if (digitalRead(START) == LOW)
  {
    delay(10); //debounce delay
    if(digitalRead(START)==LOW) //confirm the input level change
    {
      while(digitalRead(START) == LOW);
      digitalWrite(led,LOW);        //make LED off
      while(ee_addr <1024)
      {
        //--- Read the analog input of the sensor
        analog_val= analogRead(input);
        if(storage == BIT_8)
        {
          analog_val= analog_val/4;      //divide the value by 4 to make it 8
bit
        }
        sample[ee_addr]=(analog_val<<8)>>8;   //store the lower byte
        ee_addr += 1;          //ram_address address increment by one
        if(storage == BIT_10)
        {
          sample[ee_addr]= analog_val>>8;   //store the Higher byte
          ee_addr += 1;
        }

        if(time_unit == micro)      //call the delay of us if time unit is micro
        {
          delayMicroseconds((delay_val-115));
        }
        else      //call delay in milliseconds
        {
          delay(delay_val);
        }
        //          digitalWrite(led,(1^(digitalRead(led))));
      }
      ee_addr =0;      //if all the locations are written reset the ee_addr
variable
      for(ee_addr=0;ee_addr<1024;ee_addr++)
      {
        EEPROM.write(ee_addr, sample[ee_addr]);  //Store it to EEPROM
      }
```

```
      ee_addr =0;
      digitalWrite(led,HIGH); //Turn on the LED to indicate data is stored in
EEPROM
    }
  }
  //----------   Check READ key is pressed or not ------------
  if (digitalRead(READ) == LOW)
  {
    delay(10); //debounce delay
    if(digitalRead(READ) == LOW) //confirm the input level change
    {
      int adr=0;
      int int_val=0;
      byte val;
      while(adr<1024)  //read all the 1023 places
      {
        val = EEPROM.read(adr);  //read the value
        adr += 1;                //increment the address
        if(storage == BIT_10)
        {
          int_val = (EEPROM.read(adr)<<8)+val;
          adr += 1;
        }
        else
        {
          int_val = val;        //copy val to int_val as it is.
        }
        Serial.print(int_val);       //send the value to serial line
        Serial.print(",");        //separate the two values by','
      }
      while(digitalRead(READ) == LOW); //if pressed wait till release

    }
  }

}
```