**[Light-and-Temp-logger](http://github.com/adafruit/Light-and-Temp-logger/tree/master)** / lighttemplogger.pde lighttemplogger.pde 

Txt100644 211 lines (172 sloc) 5.455 kb

* [raw](http://github.com/adafruit/Light-and-Temp-logger/raw/master/lighttemplogger.pde)
* [blame](http://github.com/adafruit/Light-and-Temp-logger/blame/master/lighttemplogger.pde)
* [history](http://github.com/adafruit/Light-and-Temp-logger/commits/master/lighttemplogger.pde)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211 | #include <SdFat.h>  #include <Wire.h>  #include "RTClib.h"  // A simple data logger for the Arduino analog pins  #define LOG\_INTERVAL 1000 // mills between entries  #define ECHO\_TO\_SERIAL 1 // echo data to serial port  #define WAIT\_TO\_START 0 // Wait for serial input in setup()  #define SYNC\_INTERVAL 1000 // mills between calls to sync()  uint32\_t syncTime = 0; // time of last sync()  // the digital pins that connect to the LEDs  #define redLEDpin 2  #define greenLEDpin 3  // The analog pins that connect to the sensors  #define photocellPin 0 // analog 0  #define tempPin 1 // analog 1  #define BANDGAPREF 14 // special indicator that we want to measure the bandgap  #define aref\_voltage 3.3 // we tie 3.3V to ARef and measure it with a multimeter!  #define bandgap\_voltage 1.1 // this is not super guaranteed but its not -too- off  RTC\_DS1307 RTC; // define the Real Time Clock object  // The objects to talk to the SD card  Sd2Card card;  SdVolume volume;  SdFile root;  SdFile file;  void error(char \*str)  {    Serial.print("error: ");    Serial.println(str);    while(1);  }  void setup(void)  {    Serial.begin(9600);    Serial.println();    #if WAIT\_TO\_START    Serial.println("Type any character to start");    while (!Serial.available());  #endif //WAIT\_TO\_START    // initialize the SD card    if (!card.init()) error("card.init");      // initialize a FAT volume    if (!volume.init(card)) error("volume.init");      // open root directory    if (!root.openRoot(volume)) error("openRoot");      // create a new file    char name[] = "LOGGER00.CSV";    for (uint8\_t i = 0; i < 100; i++) {      name[6] = i/10 + '0';      name[7] = i%10 + '0';      if (file.open(root, name, O\_CREAT | O\_EXCL | O\_WRITE)) break;    }    if (!file.isOpen()) error ("file.create");    Serial.print("Logging to: ");    Serial.println(name);    // write header    file.writeError = 0;    Wire.begin();    if (!RTC.begin()) {      file.println("RTC failed");  #if ECHO\_TO\_SERIAL      Serial.println("RTC failed");  #endif //ECHO\_TO\_SERIAL    }      file.println("millis,stamp,datetime,light,temp,vcc");  #if ECHO\_TO\_SERIAL    Serial.println("millis,stamp,datetime,light,temp,vcc");  #endif //ECHO\_TO\_SERIAL    // attempt to write out the header to the file    if (file.writeError || !file.sync()) {      error("write header");    }      pinMode(redLEDpin, OUTPUT);    pinMode(greenLEDpin, OUTPUT);      // If you want to set the aref to something other than 5v    analogReference(EXTERNAL);  }  void loop(void)  {    DateTime now;      // clear print error    file.writeError = 0;    // delay for the amount of time we want between readings    delay((LOG\_INTERVAL -1) - (millis() % LOG\_INTERVAL));      digitalWrite(redLEDpin, HIGH);    // log milliseconds since starting    uint32\_t m = millis();    file.print(m); // milliseconds since start    file.print(", ");  #if ECHO\_TO\_SERIAL    Serial.print(m); // milliseconds since start    Serial.print(", ");  #endif    // fetch the time    now = RTC.now();    // log time    file.print(now.unixtime()); // seconds since 1/1/1970    file.print(", ");    file.print('"');    file.print(now.year(), DEC);    file.print("/");    file.print(now.month(), DEC);    file.print("/");    file.print(now.day(), DEC);    file.print(" ");    file.print(now.hour(), DEC);    file.print(":");    file.print(now.minute(), DEC);    file.print(":");    file.print(now.second(), DEC);    file.print('"');  #if ECHO\_TO\_SERIAL    Serial.print(now.unixtime()); // seconds since 1/1/1970    Serial.print(", ");    Serial.print('"');    Serial.print(now.year(), DEC);    Serial.print("/");    Serial.print(now.month(), DEC);    Serial.print("/");    Serial.print(now.day(), DEC);    Serial.print(" ");    Serial.print(now.hour(), DEC);    Serial.print(":");    Serial.print(now.minute(), DEC);    Serial.print(":");    Serial.print(now.second(), DEC);    Serial.print('"');  #endif //ECHO\_TO\_SERIAL    analogRead(photocellPin);    delay(10);    int photocellReading = analogRead(photocellPin);      analogRead(tempPin);    delay(10);    int tempReading = analogRead(tempPin);      // converting that reading to voltage, for 3.3v arduino use 3.3, for 5.0, use 5.0    float voltage = tempReading \* aref\_voltage / 1024;    float temperatureC = (voltage - 0.5) \* 100 ;    float temperatureF = (temperatureC \* 9 / 5) + 32;      file.print(", ");    file.print(photocellReading);    file.print(", ");    file.print(temperatureF);  #if ECHO\_TO\_SERIAL    Serial.print(", ");    Serial.print(photocellReading);    Serial.print(", ");    Serial.print(temperatureF);  #endif //ECHO\_TO\_SERIAL    // Log the estimated 'VCC' voltage by measuring the internal 1.1v ref    analogRead(BANDGAPREF);    delay(10);    int refReading = analogRead(BANDGAPREF);    float supplyvoltage = (bandgap\_voltage \* 1024) / refReading;      file.print(", ");    file.print(supplyvoltage);  #if ECHO\_TO\_SERIAL    Serial.print(", ");    Serial.print(supplyvoltage);  #endif // ECHO\_TO\_SERIAL    file.println();  #if ECHO\_TO\_SERIAL    Serial.println();  #endif // ECHO\_TO\_SERIAL    if (file.writeError) error("write data");    digitalWrite(redLEDpin, LOW);      //don't sync too often - requires 2048 bytes of I/O to SD card    if ((millis() - syncTime) < SYNC\_INTERVAL) return;    syncTime = millis();      // blink LED to show we are syncing data to the card & updating FAT!    digitalWrite(greenLEDpin, HIGH);    if (!file.sync()) error("sync");    digitalWrite(greenLEDpin, LOW);  } |