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

100644 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)

|  |  |
| --- | --- |
| 123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119120121122123124125126127128129130131132133134135136137138139140141142143144145146147148149150151152153154155156157158159160161162163164165166167168169170171172173174175176177178179180181182183184185186187188189190191192193194195196197198199200201202203204205206207208209210211 | #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- offRTC\_DS1307 RTC; // define the Real Time Clock object// The objects to talk to the SD cardSd2Card 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);} |