

[Light-and-Temp-logger](#) / lighttemplogger.pde

100644 211 lines (172 sloc) 5.455 kb

- [raw](#)
- [blame](#)
- [history](#)

```
1 #include <SdFat.h>
2 #include <Wire.h>
3 #include "RTCLib.h"
4
5 // A simple data logger for the Arduino analog pins
6 #define LOG_INTERVAL 1000 // mills between entries
7 #define ECHO_TO_SERIAL 1 // echo data to serial port
8 #define WAIT_TO_START 0 // Wait for serial input in setup()
9 #define SYNC_INTERVAL 1000 // mills between calls to sync()
10 uint32_t syncTime = 0; // time of last sync()
11
12 // the digital pins that connect to the LEDs
13 #define redLEDPin 2
14 #define greenLEDPin 3
15
16 // The analog pins that connect to the sensors
17 #define photocellPin 0 // analog 0
18 #define tempPin 1 // analog 1
19 #define BANDGAPREF 14 // special indicator that we want to measure the
20 bandgap
21
22 #define aref_voltage 3.3 // we tie 3.3V to ARef and measure it with a
23 multimeter!
24 #define bandgap_voltage 1.1 // this is not super guaranteed but its not -too- off
25
26 RTC_DS1307 RTC; // define the Real Time Clock object
27
28 // The objects to talk to the SD card
29 Sd2Card card;
30 SdVolume volume;
31 SdFile root;
32 SdFile file;
33
34 void error(char *str)
35 {
36   Serial.print("error: ");
37   Serial.println(str);
38   while(1);
39 }
40
41 void setup(void)
42 {
43   Serial.begin(9600);
44   Serial.println();
45
46   #if WAIT_TO_START
47     Serial.println("Type any character to start");
48     while (!Serial.available());
49   #endif //WAIT_TO_START
50
51   // initialize the SD card
52   if (!card.init()) error("card.init");
53
54   // initialize a FAT volume
```

```

55  if (!volume.init(card)) error("volume.init");
56
57  // open root directory
58  if (!root.openRoot(volume)) error("openRoot");
59
60  // create a new file
61  char name[] = "LOGGER00.CSV";
62  for (uint8_t i = 0; i < 100; i++) {
63      name[6] = i/10 + '0';
64      name[7] = i%10 + '0';
65      if (file.open(root, name, O_CREAT | O_EXCL | O_WRITE)) break;
66  }
67  if (!file.isOpen()) error ("file.create");
68  Serial.print("Logging to: ");
69  Serial.println(name);
70
71  // write header
72  file.writeError = 0;
73
74  Wire.begin();
75  if (!RTC.begin()) {
76      file.println("RTC failed");
77  #if ECHO_TO_SERIAL
78      Serial.println("RTC failed");
79  #endif //ECHO_TO_SERIAL
80  }
81
82
83  file.println("millis,stamp,datetime,light,temp,vcc");
84  #if ECHO_TO_SERIAL
85      Serial.println("millis,stamp,datetime,light,temp,vcc");
86  #endif //ECHO_TO_SERIAL
87
88  // attempt to write out the header to the file
89  if (file.writeError || !file.sync()) {
90      error("write header");
91  }
92
93  pinMode(redLEDPin, OUTPUT);
94  pinMode(greenLEDPin, OUTPUT);
95
96  // If you want to set the aref to something other than 5v
97  analogReference(EXTERNAL);
98 }
99
100void loop(void)
101{
102  DateTime now;
103
104  // clear print error
105  file.writeError = 0;
106
107  // delay for the amount of time we want between readings
108  delay((LOG_INTERVAL -1) - (millis() % LOG_INTERVAL));
109
110  digitalWrite(redLEDPin, HIGH);
111
112  // log milliseconds since starting
113  uint32_t m = millis();
114  file.print(m);           // milliseconds since start
115  file.print(", ");
116  #if ECHO_TO_SERIAL
117  Serial.print(m);       // milliseconds since start

```

```

118 Serial.print(", ");
119#endif
120
121 // fetch the time
122 now = RTC.now();
123 // log time
124 file.print(now.unixtime()); // seconds since 1/1/1970
125 file.print(", ");
126 file.print('');
127 file.print(now.year(), DEC);
128 file.print("/");
129 file.print(now.month(), DEC);
130 file.print("/");
131 file.print(now.day(), DEC);
132 file.print(" ");
133 file.print(now.hour(), DEC);
134 file.print(":");
135 file.print(now.minute(), DEC);
136 file.print(":");
137 file.print(now.second(), DEC);
138 file.print('');
139#if ECHO_TO_SERIAL
140 Serial.print(now.unixtime()); // seconds since 1/1/1970
141 Serial.print(", ");
142 Serial.print('');
143 Serial.print(now.year(), DEC);
144 Serial.print("/");
145 Serial.print(now.month(), DEC);
146 Serial.print("/");
147 Serial.print(now.day(), DEC);
148 Serial.print(" ");
149 Serial.print(now.hour(), DEC);
150 Serial.print(":");
151 Serial.print(now.minute(), DEC);
152 Serial.print(":");
153 Serial.print(now.second(), DEC);
154 Serial.print('');
155#endif //ECHO_TO_SERIAL
156
157 analogRead(photocellPin);
158 delay(10);
159 int photocellReading = analogRead(photocellPin);
160
161 analogRead(tempPin);
162 delay(10);
163 int tempReading = analogRead(tempPin);
164
165 // converting that reading to voltage, for 3.3v arduino use 3.3, for 5.0, use 5.0
166 float voltage = tempReading * aref_voltage / 1024;
167 float temperatureC = (voltage - 0.5) * 100 ;
168 float temperatureF = (temperatureC * 9 / 5) + 32;
169
170 file.print(", ");
171 file.print(photocellReading);
172 file.print(", ");
173 file.print(temperatureF);
174#if ECHO_TO_SERIAL
175 Serial.print(", ");
176 Serial.print(photocellReading);
177 Serial.print(", ");
178 Serial.print(temperatureF);
179#endif //ECHO_TO_SERIAL
180

```

```
181 // Log the estimated 'VCC' voltage by measuring the internal 1.1v ref
182 analogRead(BANDGAPREF);
183 delay(10);
184 int refReading = analogRead(BANDGAPREF);
185 float supplyvoltage = (bandgap_voltage * 1024) / refReading;
186
187 file.print(", ");
188 file.print(supplyvoltage);
189#if ECHO_TO_SERIAL
190 Serial.print(", ");
191 Serial.print(supplyvoltage);
192#endif // ECHO_TO_SERIAL
193
194 file.println();
195#if ECHO_TO_SERIAL
196 Serial.println();
197#endif // ECHO_TO_SERIAL
198
199 if (file.writeError) error("write data");
200 digitalWrite(redLEDpin, LOW);
201
202 //don't sync too often - requires 2048 bytes of I/O to SD card
203 if ((millis() - syncTime) < SYNC_INTERVAL) return;
204 syncTime = millis();
205
206 // blink LED to show we are syncing data to the card & updating FAT!
207 digitalWrite(greenLEDpin, HIGH);
208 if (!file.sync()) error("sync");
209 digitalWrite(greenLEDpin, LOW);
210}
211
```