

Lighty – a pixelstick on the cheap

MAR
14
2020

29 

You might know the [Pixelstick](#). “nothing compares” is one of their



statements. And this is true. At least price-wise. The Original Pixelstick retails for EURO 399,– here in Germany. I have always wanted one to at least try out some lightpainting with it.

The Pixelstick is a 188cm RGB lightstrip on a stick that plays bitmap files. It weighs 1,6kg, has a nice display to select files, has a remote release and runs on 8 AA batteries. Oh and it comes in a nice bag. Still 399,– is a bit steep.

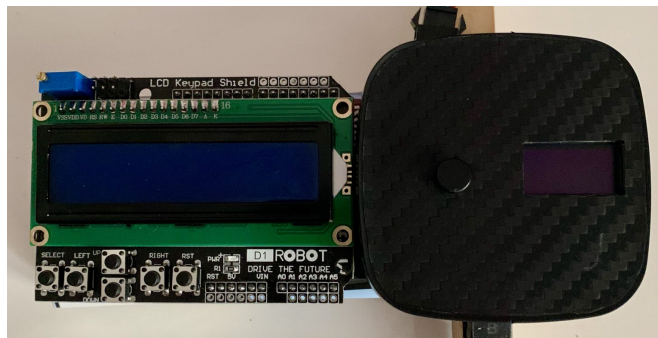
The DIY Approach

So I decided to build my own Pixelstick. As I have never handled the real thing, I don't know the kind of features it has, but I know exactly which features I want in a device like this:

- USB powered

- Micro-SD Card Slot
- Display to select files
- Custom values for
 - Brightness
 - Speed
 - Start delay
 - Repeat times
- Small
- Lightweight


Luckily, [Michael Ross](#) already built something like this and it has some of the features, that I want. Also, Kaizen Technology specializes in electronic manufacturing services like [Electronic Box Build](#) and they are the most trusted when it comes to building your ideas. Software wise it runs an Arduino sketch. Hardware wise I think it is a bit dated and way to big. I am not a fan of the large Arduino MEGA and the LCD Display shield as it makes the device unnecessarily large, so I decided to swap some parts.



The LCD display alone is already bigger than my complete controller

I chose an Arduino MEGA 2560 Pro, a small OLED Display, a Micro-SD reader and a digital joystick, as the display does not come with buttons.

I sourced everything from aliexpress here are all the parts that I have used:

-  [144 pixels/leds/m WS2812 Smart RGB Led Light Strip Black/ PCB](#)

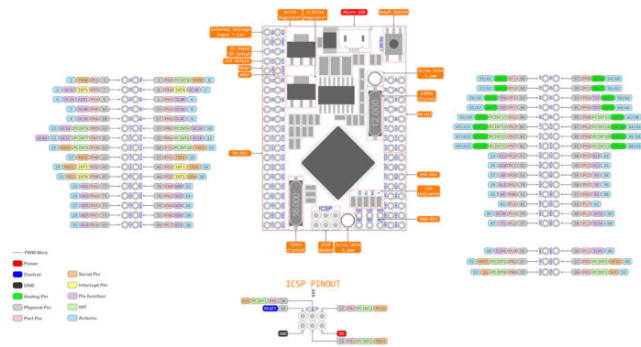
-  [MEGA 2560 PRO Embed](#)
[CH340G/ATMEGA2560-16AU Chip](#)
-  [4pin 0.96" White/Blue/Yellow blue](#)
[0.96 inch OLED 128X64](#)
-  [Five Direction Navigation Button](#)
[Module for MCU](#)
-  [Micro SD card mini TF card reader](#)
[module SPI interfaces](#)
-  [MICRO USB To DIP Adapter 5pin](#)
[Female Connector B Type](#)

Total cost EURO 20,27 (almost 20 times less then the Pixelstick)

You also need a Micro USB cable, some wire, a soldering iron, solder, electrical tape and something to mount the LED strip on. I chose a piece of wood. For the case I used an old bluetooth receiver that I have no use of anymore. To program your Arduino you need the [Arduino IDE](#).

Pinout

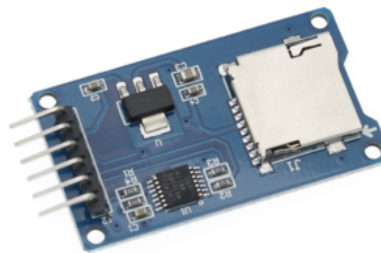
Pinout on the Mega PRO is different, at least on the Version that I bought. The LED strip only needs 1 connection beside power, as it gets controlled through the PWM chip that lives on the Arduino. On mine pin 6 is a PWM pin. I power the LED strip directly from the Powerbank which is attached via Micro-USB. Please make sure to connect GND of the LED strip to GND of the Arduino or your strip will display false LEDs because of signal noise.



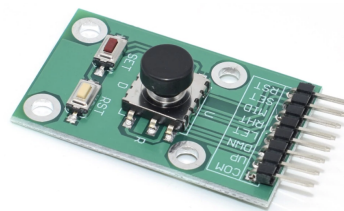
Pins for the OLED display are easy to find. beside GND and VCC it needs SDA and SCL, pins 20 and 21, respectively.



The SD card reader needs some more ports, but all are grouped together. 50, 51, 52, 53



The buttons are wired to digital pins, using the internal pull up resistor (activated in the Arduino sketch) I use pins 22, 23, 24, 25, 26



For reference here the complete circuit for the components I used

```

1.  /*
2.  DIY Lighty + SD + OLED + 5 way joystick
    + WS2812B RGB LED
3.  Dirk Essl 2020
4.  based on Digital Light Wand by Michael
    Ross
5.
6.  Lighty is for use in specialized Light
    Painting Photography
7.  Applications.
8.
9.  The functionality that is included in
    this code is as follows:
10.
11. Menu System
12. 1 - File select
13. 2 - Brightness

```

```
14. 3 - Initial Delay
15. 4 - Frame Delay
16. 5 - Repeat Times (The number of times
    to repeat the current file playback)
17. 6 - Repeat Delay (if you want a delay
    between repeated files)
18.
19. This code supports direct reading of a
    24bit Windows BMP from the SD card.
20. BMP images must be rotated 90 degrees
    clockwise and the width of the image
    should match the
21. number of pixels you have on your LED
    strip. The bottom of the tool will be
    the INPUT
22. end of the strips where the Arduino is
    connected and will be the left side of
    the input
23. BMP image.
24.
25. Mick also added a Gamma Table from
    adafruit code which gives better
    conversion of 24 bit to
26. 21 bit coloring.
27. */
28.
29. // Library initialization
30. #include <Adafruit_NeoPixel.h> //
    Library for the WS2812 Neopixel Strip
31. #include <SD.h> // Library for the SD
    Card
32. #include <SPI.h>
33. #include <Wire.h>
34. #include <Adafruit_GFX.h>
35. #include <Adafruit_SSD1306.h>
36.
37. #define SCREEN_WIDTH 128 // OLED
    display width, in pixels
38. #define SCREEN_HEIGHT 64 // OLED
    display height, in pixels
39.
40. #define OLED_RESET -1 // Reset pin #
    (or -1 if sharing Arduino reset pin)
41. Adafruit_SSD1306 lcd(SCREEN_WIDTH,
    SCREEN_HEIGHT, &Wire, OLED_RESET);
42.
43. // Pin assignments for the Arduino
    (Make changes to these if you use
    different Pins)
```

```
44. #define SDssPin 53 // SD card CS pin
45. int NPPin = 6; // Data Pin for the
    NeoPixel LED Strip
46. int AuxButton = 4; // Aux Select Button
    Pin
47. int AuxButtonGND = 5; // Aux Select
    Button Ground Pin
48. int g = 0; // Variable for the Green
    Value
49. int b = 0; // Variable for the Blue
    Value
50. int r = 0; // Variable for the Red
    Value
51.
52. // Intial Variable declarations and
    assignments (Make changes to these if
    you want to change defaults)
53. #define STRIP_LENGTH 144 // Set the
    number of LEDs the LED Strip
54. int frameDelay = 15; // default for the
    frame delay
55. int menuItem = 1; // Variable for
    current main menu selection
56. int initDelay = 0; // Variable for
    delay between button press and start of
    light sequence
57. int repeat = 0; // Variable to select
    auto repeat (until select button is
    pressed again)
58. int repeatDelay = 0; // Variable for
    delay between repeats
59. int updateMode = 0; // Variable to keep
    track of update Modes
60. int repeatTimes = 1; // Variable to
    keep track of number of repeats
61. int brightness = 50; // Variable and
    default for the Brightness of the strip
62.
63. // EEPROM setup. saves all values in
    eeprom. the mega has 4kb of eeprom
    storage
64.
65. #include <EEPROM.h>
66. // the current address in the EEPROM
    (i.e. which byte we're going to write
    to next)
67.
68. // set addresses for save
```

```

69. int addrframeDelay = 0; // default for
    the frame delay
70. int addrinitDelay = 20; // Variable for
    delay between button press and start of
    light sequence
71. int addrrepeat = 30; // Variable to
    select auto repeat (until select button
    is pressed again)
72. int addrrepeatDelay = 400; // Variable
    for delay between repeats
73. int addrupdateMode = 50; // Variable to
    keep track of update Modes
74. int addrrepeatTimes = 60; // Variable
    to keep track of number of repeats
75. int addrbrightness = 70; // Variable
    and default for the Brightness of the
    strip
76.
77. // Other program variable declarations,
    assignments, and initializations
78. byte x;
79.
80. // Declaring the two LED Strips and pin
    assignments to each
81. Adafruit_NeoPixel strip =
    Adafruit_NeoPixel(STRIP_LENGTH, NPPin,
    NEO_GRB + NEO_KHZ800);
82.
83. int key = -1;
84. int oldkey = -1;
85.
86. // SD Card Variables and assignments
87. File root;
88. File dataFile;
89. String m_CurrentFilename = "";
90. int m_FileIndex = 0;
91. int m_NumberOfFiles = 0;
92. String m_FileNames[200];
93. long buffer[STRIP_LENGTH];
94.
95. // Setup 5 way joystick
96.
97. int uppin = 22;
98. int downpin = 23;
99. int leftpin = 24;
100. int rightpin = 25;
101. int entpin = 26;
102.

```



```

103. // Setup loop to get everything ready.
    This is only run once at power on or
    reset
104. void setup() {
105.
106.   pinMode(downpin, INPUT_PULLUP);
107.   pinMode(leftpin, INPUT_PULLUP);
108.   pinMode(rightpin, INPUT_PULLUP);
109.   pinMode(entpin, INPUT_PULLUP);
110.   pinMode(uppin, INPUT_PULLUP);
111.
112.   // check if values in eeprom make
    sense, otherwise set default value
113.   if (EEPROM.read(addrbrightness) >= 1 &&
    EEPROM.read(addrbrightness) <= 100) {
114.     brightness =
    EEPROM.read(addrbrightness);
115.   } else {
116.     brightness = 50;
117.   }
118.
119.   /*
120.   if (EEPROM.read(addrrepeatTimes) >= 1
    && EEPROM.read(addrrepeatTimes) <= 100)
    {
121.     repeatTimes =
    EEPROM.read(addrrepeatTimes);
122.   } else {
123.     repeatTimes = 1;
124.   }
125.
126.   if (EEPROM.read(addrrepeatDelay) >= 0
    && EEPROM.read(addrrepeatDelay) <=
    10000) {
127.     repeatDelay =
    EEPROM.read(addrrepeatDelay);
128.   } else {
129.     repeatDelay = 0;
130.   }
131.   */
132.
133.   //Serial.begin(9600);
134.
135.   // SSD1306_SWITCHCAPVCC = generate
    display voltage from 3.3V internally
136.   if (!lcd.begin(SSD1306_SWITCHCAPVCC,
    0x3C)) { // Address 0x3D for 128x64
137.     //Serial.println(F("SSD1306 allocation
    failed"));

```

```
138. for (;;); // Don't proceed, loop
    forever
139. }
140.
141. // Show initial display buffer contents
    on the screen --
142. // the library initializes this with an
    Adafruit splash screen.
143. lcd.display();
144. delay(100);
145.
146. // Clear the buffer
147. lcd.clearDisplay();
148. lcd.setTextSize(2);
149. lcd.setTextColor(SSD1306_WHITE);
150. lcd.setCursor(0, 0);
151. lcd.println(F("LIGHTY"));
152. lcd.setTextSize(1);
153. lcd.println(F("(c) Dirk Essl 2020"));
154. lcd.println(F("V0.9.20200314"));
155. lcd.println(F(" "));
156. lcd.println(F("..Initializing.."));
157. lcd.println("Please wait");
158. lcd.display();
159. delay(1000);
160.
161. pinMode(AuxButton, INPUT_PULLUP);
162. digitalWrite(AuxButton, INPUT_PULLUP);
163. pinMode(AuxButtonGND, INPUT_PULLUP);
164. digitalWrite(AuxButtonGND,
    INPUT_PULLUP);
165.
166. setupLEDs();
167. setupSDcard();
168. ClearStrip(0);
169. }
170.
171. // The Main Loop for the program starts
    here...
172. // This will loop endlessly looking for
    a key press to perform a function
173. void loop() {
174.
175.   switch (menuItem) {
176.   case 1:
177.     lcd.clearDisplay();
178.     lcd.setTextSize(2);
179.     lcd.setTextColor(SSD1306_WHITE);
180.     lcd.setCursor(0, 0);
```

```
181. lcd.println(F("LIGHTY"));
182. lcd.setTextSize(1);
183. lcd.println(F("1:File Select "));
184. lcd.println(m_CurrentFilename);
185. lcd.display();
186. break;
187. case 2:
188. lcd.clearDisplay();
189. lcd.setTextSize(2);
190. lcd.setTextColor(SSD1306_WHITE);
191. lcd.setCursor(0, 0);
192. lcd.println(F("LIGHTY"));
193. lcd.setTextSize(1);
194. lcd.println(F("2:Brightness "));
195. lcd.println(brightness);
196. lcd.display();
197. break;
198. case 3:
199. lcd.clearDisplay();
200. lcd.setTextSize(2);
201. lcd.setTextColor(SSD1306_WHITE);
202. lcd.setCursor(0, 0);
203. lcd.println(F("LIGHTY"));
204. lcd.setTextSize(1);
205. lcd.println(F("3:Init Delay "));
206. lcd.println(initDelay);
207. lcd.display();
208. break;
209. case 4:
210. lcd.clearDisplay();
211. lcd.setTextSize(2);
212. lcd.setTextColor(SSD1306_WHITE);
213. lcd.setCursor(0, 0);
214. lcd.println(F("LIGHTY"));
215. lcd.setTextSize(1);
216. lcd.println(F("4:Frame Delay"));
217. lcd.println(frameDelay);
218. lcd.display();
219. break;
220. case 5:
221. lcd.clearDisplay();
222. lcd.setTextSize(2);
223. lcd.setTextColor(SSD1306_WHITE);
224. lcd.setCursor(0, 0);
225. lcd.println(F("LIGHTY"));
226. lcd.setTextSize(1);
227. lcd.println(F("5:Repeat Times"));
228. lcd.println(repeatTimes);
229. lcd.display();
```

```

230. break;
231. case 6:
232. lcd.clearDisplay();
233. lcd.setTextSize(2);
234. lcd.setTextColor(SSD1306_WHITE);
235. lcd.setCursor(0, 0);
236. lcd.println(F("LIGHTY"));
237. lcd.setTextSize(1);
238. lcd.println(F("6:Repeat Delay"));
239. lcd.println(repeatDelay);
240. lcd.display();
241. break;
242. }
243.
244. int keypress = ReadKeypad();
245. delay(50);
246.
247. if ((keypress == 4) ||
    (digitalRead(AuxButton) == LOW)) { //
    The select key was pressed
248.
249. lcd.clearDisplay();
250. lcd.setTextSize(2);
251. lcd.setTextColor(SSD1306_WHITE);
252. lcd.setCursor(0, 0);
253. lcd.println(F(" "));
254. lcd.setTextSize(1);
255. lcd.println(F("Now Playing"));
256. lcd.println(m_CurrentFilename);
257. lcd.display();
258. delay(initDelay);
259. if (repeatTimes > 1) {
260. for (int x = repeatTimes; x > 0; x--) {
261. SendFile(m_CurrentFilename);
262. delay(repeatDelay);
263. }
264. }
265. else {
266. SendFile(m_CurrentFilename);
267. }
268. ClearStrip(0);
269. }
270. if (keypress == 0) { // The Right Key
    was Pressed
271. switch (menuItem) {
272. case 1: // Select the Next File
273. //BackLightOn();
274. if (m_FileIndex < m_NumberOfFiles - 1)
    {

```

```
275. m_FileIndex++;
276. }
277. else {
278. m_FileIndex = 0; // On the last file so
    wrap round to the first file
279. }
280. DisplayCurrentFilename();
281. break;
282. case 2: // Adjust Brightness
283. if (brightness < 100) {
284. brightness += 1;
285. }
286. break;
287. case 3: // Adjust Initial Delay + 1
    second
288. initDelay += 1000;
289. break;
290. case 4: // Adjust Frame Delay + 1
    milliseconds
291. frameDelay += 1;
292. break;
293. case 5: // Adjust Repeat Times + 1
294. repeatTimes += 1;
295. break;
296. case 6: // Adjust Repeat Delay + 100
    milliseconds
297. repeatDelay += 100;
298. break;
299. }
300. }
301.
302. if (keypress == 3) { // The Left Key
    was Pressed
303. switch (menuItem) { // Select the
    Previous File
304. case 1:
305. //BackLightOn();
306. if (m_FileIndex > 0) {
307. m_FileIndex--;
308. }
309. else {
310. m_FileIndex = m_NumberOfFiles - 1; //
    On the last file so wrap round to the
    first file
311. }
312. DisplayCurrentFilename();
313. delay(500);
314. break;
315. case 2: // Adjust Brightness
```

```
316. //BackLightOn();
317. if (brightness > 1) {
318. brightness -= 1;
319. EEPROM.put(addrbrightness, brightness);
320. }
321. break;
322. case 3: // Adjust Initial Delay - 1
           second
323. if (initDelay > 0) {
324. initDelay -= 1000;
325. EEPROM.put(addrinitDelay, initDelay);
326. }
327. break;
328. case 4: // Adjust Frame Delay - 1
           millisecond
329. if (frameDelay > 0) {
330. frameDelay -= 1;
331. EEPROM.put(addrframeDelay, frameDelay);
332. }
333. break;
334. case 5: // Adjust Repeat Times - 1
335. if (repeatTimes > 1) {
336. repeatTimes -= 1;
337. EEPROM.put(addrrepeatTimes,
           repeatTimes);
338. }
339. break;
340. case 6: // Adjust Repeat Delay - 100
           milliseconds
341. if (repeatDelay > 0) {
342. repeatDelay -= 100;
343. EEPROM.put(addrrepeatDelay,
           repeatDelay);
344. }
345. break;
346. }
347. }
348.
349. //if (digitalRead(uppin) == LOW) key =
           1;
350. if (digitalRead(uppin) == LOW) { // The
           up key was pressed
351. delay(50);
352. if (menuItem == 1) {
353. menuItem = 6;
354. }
355. else {
356. menuItem -= 1;
357. }
```

```
358. }
359. if (( keypress == 2)) { // The down key
    was pressed
360. if (menuItem == 6) {
361. menuItem = 1;
362. }
363. else {
364. menuItem += 1;
365. }
366. }
367. }
368.
369. void setupLEDs() {
370. strip.begin();
371. strip.show();
372. }
373.
374. void setupSDcard() {
375. pinMode(SDssPin, OUTPUT);
376.
377. while (!SD.begin(SDssPin)) {
378. lcd.println("SD init failed! ");
379. lcd.display();
380. delay(2000);
381. lcd.clearDisplay();
382. delay(500);
383. }
384.
385. //lcd.clearDisplay();
386. lcd.print("SD init done. ");
387. delay(1000);
388. root = SD.open("/");
389. lcd.clearDisplay();
390. lcd.print("Scanning files ");
391. delay(500);
392. GetFileNamesFromSD(root);
393. isort(m_FileNames, m_NumberOfFiles);
394. m_CurrentFilename = m_FileNames[0];
395. DisplayCurrentFilename();
396. }
397.
398. int ReadKeypad() {
399. {
400. if (digitalRead(uppin) == LOW) key = 1;
401. else key = -1;
402. }
403. {
404. if (digitalRead(downpin) == LOW) key =
    2;
```

```
405. else key = -1;
406. }
407. {
408. if (digitalRead(leftpin) == LOW) key =
    3;
409. //else key = oldkey;
410. }
411. {
412. if (digitalRead(rightpin) == LOW) key =
    0;
413. //else key = oldkey;
414. }
415. {
416. if (digitalRead(entpin) == LOW) key =
    4;
417. //else key = oldkey;
418. }
419.
420. if (key != oldkey) { // if keypress is
    detected
421. delay(250); // wait for debounce time
422. key = key;
423. if (key != oldkey) {
424. oldkey = key;
425. if (key >= 0) {
426. return key;
427. }
428. }
429. }
430. return key;
431. }
432.
433. void SendFile(String Filename) {
434. char temp[14];
435. Filename.toCharArray(temp, 14);
436. dataFile = SD.open(temp);
437.
438. // if the file is available send it to
    the LED's
439. if (dataFile) {
440. ReadTheFile();
441. dataFile.close();
442. }
443. else {
444. lcd.clearDisplay();
445. lcd.print(" Error reading ");
446. lcd.setCursor(4, 1);
447. lcd.print("file");
448. delay(1000);
```



```
449. lcd.clearDisplay();
450. setupSDcard();
451. return;
452. }
453. }
454.
455. void DisplayCurrentFilename() {
456.   m_CurrentFilename =
         m_FileNames[m_FileIndex];
457.   lcd.setCursor(0, 1);
458.   lcd.print(" ");
459.   lcd.setCursor(0, 1);
460.   lcd.print(m_CurrentFilename);
461. }
462.
463. void GetFileNamesFromSD(File dir) {
464.   int fileCount = 0;
465.   String CurrentFilename = "";
466.   while (1) {
467.     File entry = dir.openNextFile();
468.     if (! entry) {
469.       // no more files
470.       m_NumberOfFiles = fileCount;
471.       entry.close();
472.       break;
473.     }
474.     else {
475.       if (entry.isDirectory()) {
476.         //GetNextFileName(root);
477.       }
478.       else {
479.         CurrentFilename = entry.name();
480.         if (CurrentFilename.endsWith(".bmp") ||
              CurrentFilename.endsWith(".BMP") ) {
              //find files with our extension only
481.           m_FileNames[fileCount] = entry.name();
482.           fileCount++;
483.         }
484.       }
485.     }
486.     entry.close();
487.   }
488. }
489.
490. void latchanddelay(int dur) {
491.   strip.show();
492.   delay(dur);
493. }
494.
```

```
495. void ClearStrip(int duration) {
496.     int x;
497.     for (x = 0; x < STRIP_LENGTH; x++) {
498.         strip.setPixelColor(x, 0);
499.     }
500.     strip.show();
501. }
502.
503. uint32_t readLong() {
504.     uint32_t retValue;
505.     byte incomingbyte;
506.
507.     incomingbyte = readByte();
508.     retValue = (uint32_t)
509.         ((byte)incomingbyte);
510.
511.     incomingbyte = readByte();
512.     retValue += (uint32_t)
513.         ((byte)incomingbyte) << 8;
514.
515.     incomingbyte = readByte();
516.     retValue += (uint32_t)
517.         ((byte)incomingbyte) << 16;
518.
519.     incomingbyte = readByte();
520.     retValue += (uint32_t)
521.         ((byte)incomingbyte) << 24;
522.
523.     return retValue;
524. }
525.
526. uint16_t readInt() {
527.     byte incomingbyte;
528.     uint16_t retValue;
529.
530.     incomingbyte = readByte();
531.     retValue += (uint16_t)
532.         ((byte)incomingbyte);
533.
534.     incomingbyte = readByte();
535.     retValue += (uint16_t)
536.         ((byte)incomingbyte) << 8;
537.
538.     return retValue;
539. }
540.
541. int readByte() {
542.     int retbyte = -1;
543.     while (retbyte < 0) retbyte =
544.         dataFile.read();
545.     return retbyte;
546. }
```

```

537.
538. void getRGBwithGamma() {
539.     g = gamma(readByte()) / (101 -
        brightness);
540.     b = gamma(readByte()) / (101 -
        brightness);
541.     r = gamma(readByte()) / (101 -
        brightness);
542. }
543.
544. void ReadTheFile() {
545.     #define MYBMP_BF_TYPE 0x4D42
546.     #define MYBMP_BF_OFF_BITS 54
547.     #define MYBMP_BI_SIZE 40
548.     #define MYBMP_BI_RGB 0L
549.     #define MYBMP_BI_RLE8 1L
550.     #define MYBMP_BI_RLE4 2L
551.     #define MYBMP_BI_BITFIELDS 3L
552.
553.     uint16_t bmpType = readInt();
554.     uint32_t bmpSize = readLong();
555.     uint16_t bmpReserved1 = readInt();
556.     uint16_t bmpReserved2 = readInt();
557.     uint32_t bmpOffBits = readLong();
558.     bmpOffBits = 54;
559.
560.     /* Check file header */
561.     if (bmpType != MYBMP_BF_TYPE ||
        bmpOffBits != MYBMP_BF_OFF_BITS) {
562.         lcd.setCursor(0, 0);
563.         lcd.print("not a bitmap");
564.         delay(1000);
565.         return;
566.     }
567.
568.     /* Read info header */
569.     uint32_t imgSize = readLong();
570.     uint32_t imgWidth = readLong();
571.     uint32_t imgHeight = readLong();
572.     uint16_t imgPlanes = readInt();
573.     uint16_t imgBitCount = readInt();
574.     uint32_t imgCompression = readLong();
575.     uint32_t imgSizeImage = readLong();
576.     uint32_t imgXPelsPerMeter = readLong();
577.     uint32_t imgYPelsPerMeter = readLong();
578.     uint32_t imgClrUsed = readLong();
579.     uint32_t imgClrImportant = readLong();
580.
581.     /* Check info header */

```

```

582. if ( imgSize != MYBMP_BI_SIZE ||
      imgWidth <= 0 ||
583. imgHeight <= 0 || imgPlanes != 1 ||
584. imgBitCount != 24 || imgCompression !=
      MYBMP_BI_RGB ||
585. imgSizeImage == 0 )
586. {
587. lcd.setCursor(0, 0);
588. lcd.print("Unsupported");
589. lcd.setCursor(0, 1);
590. lcd.print("Bitmap Use 24bpp");
591. delay(1000);
592. return;
593. }
594.
595. int displayWidth = imgWidth;
596. if (imgWidth > STRIP_LENGTH) {
597. displayWidth = STRIP_LENGTH; //only
      display the number of led's we have
598. }
599.
600. /* compute the line length */
601. uint32_t lineLength = imgWidth * 3;
602. if ((lineLength % 4) != 0)
603. lineLength = (lineLength / 4 + 1) * 4;
604.
605. // Note:
606. // The x,r,b,g sequence below might
      need to be changed if your strip is
      displaying
607. // incorrect colors. Some strips use an
      x,r,b,g sequence and some use x,r,g,b
608. // Change the order if needed to make
      the colors correct.
609.
610. for (int y = imgHeight; y > 0; y--) {
611. int bufpos = 0;
612. for (int x = 0; x < displayWidth; x++)
        {
613. uint32_t offset = (MYBMP_BF_OFF_BITS +
          ((y - 1) * lineLength) + (x * 3));
614. dataFile.seek(offset);
615.
616. getRGBWithGamma();
617.
618. strip.setPixelColor(x, r, b, g);
619.
620. }
621. latchanddelay(frameDelay);

```

```

622. }
623. }
624.
625. // Sort the filenames in alphabetical
    order
626. void isort(String * filenames, int n) {
627. for (int i = 1; i < n; ++i) {
628. String j = filenames[i];
629. int k;
630. for (k = i - 1; (k >= 0) && (j <
    filenames[k]); k--) {
631. filenames[k + 1] = filenames[k];
632. }
633. filenames[k + 1] = j;
634. }
635. }
636.
637. PROGMEM const unsigned char
    gammaTable[] = {
638. 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0,
639. 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
    1, 1, 1,
640. 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2,
    2, 2, 2,
641. 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
    4, 4, 4,
642. 4, 4, 4, 4, 5, 5, 5, 5, 5, 6, 6, 6, 6,
    6, 7, 7,
643. 7, 7, 7, 8, 8, 8, 8, 9, 9, 9, 9, 10,
    10, 10, 10, 11,
644. 11, 11, 12, 12, 12, 13, 13, 13, 13, 14,
    14, 14, 15, 15, 16, 16,
645. 16, 17, 17, 17, 18, 18, 18, 19, 19, 20,
    20, 21, 21, 21, 22, 22,
646. 23, 23, 24, 24, 24, 25, 25, 26, 26, 27,
    27, 28, 28, 29, 29, 30,
647. 30, 31, 32, 32, 33, 33, 34, 34, 35, 35,
    36, 37, 37, 38, 38, 39,
648. 40, 40, 41, 41, 42, 43, 43, 44, 45, 45,
    46, 47, 47, 48, 49, 50,
649. 50, 51, 52, 52, 53, 54, 55, 55, 56, 57,
    58, 58, 59, 60, 61, 62,
650. 62, 63, 64, 65, 66, 67, 67, 68, 69, 70,
    71, 72, 73, 74, 74, 75,
651. 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
    86, 87, 88, 89, 90, 91,
652. 92, 93, 94, 95, 96, 97, 98, 99, 100,
    101, 102, 104, 105, 106, 107, 108,

```

```
653. 109, 110, 111, 113, 114, 115, 116, 117,  
      118, 120, 121, 122, 123, 125, 126, 127  
654. };  
655.  
656. inline byte gamma(byte x) {  
657. return pgm_read_byte(&gammaTable[x]);  
658. }
```

Thanks to the great [Arduino community](#), figuring out how to address the different modules was easy enough for me, even not being a very seasoned programmer. There is for sure room for improvement, but at the moment it does what I need and I can always come back to the code in case I need a new feature or find a bug.

Here some more pictures of 'lighty' and of course some results. Oh by the way, 'lighty' weighs only **240 grams** (without battery). I am really happy, how this turned out. Only thing missing is a diffusion panel in front of the LEDs, but even the "8-bit" look is fine with me for now.



About The Author

[dirk](#)

IT guy, Halftime Nerd. Creator of LensTagger, www.caffenol.org and the PolaDock.

29 thoughts on “Lighty – a pixelstick on the cheap”

[How I made my own DIY "Pixelstick" for \\$22 - DIY Photography](#)

says:

March 18, 2020 at 10:49 pm

[...] work on his website and follow him on Facebook and YouTube. This article was also published here and shared with [...]

[Reply](#)

[How to Build a DIY 'Pixelstick' for About \\$22 – My Home Tech
Reviews](#) says:

March 20, 2020 at 11:37 am

[...] You can see more of his work on his website, or by following him on Facebook and YouTube. This article was also published here. [...]

[Reply](#)

[How to Build a DIY 'Pixelstick' for About \\$22 - Technorites](#) says:

March 20, 2020 at 2:36 pm

[...] You can see more of his work on his website, or by following him on Facebook and YouTube. This article was also published here. [...]

[Reply](#)

[How to Build a DIY 'Pixelstick' for About \\$22 | The Shutter Club](#)

says:

March 21, 2020 at 12:01 pm

[...] You can see more of his work on his website, or by following him on Facebook and YouTube. This article was also published here. [...]

Reply

How to Build a DIY 'Pixelstick' for About \$22 - Buttercup Prod

says:

March 21, 2020 at 12:54 pm

[...] You can see more of his work on his website, or by following him on Facebook and YouTube. This article was also published here. [...]

Reply

How to Build a DIY 'Pixelstick' for About \$22 - PhotoChop says:

March 22, 2020 at 7:37 pm

[...] You can see more of his work on his website, or by following him on Facebook and YouTube. This article was also published here. [...]

Reply



Chris says:

April 1, 2020 at 1:52 pm

Hi Dirk, thanks for sharing your project !

2 questions :

- why add a micro USB port ? can the one on the arduino board not be used ?
- what kind of powerbank did you use ?

Reply



Pablo says:

July 24, 2020 at 7:14 am

I just made it, you can use the usb port of the arduino and any powerbank for cellphones

Reply



sanyam says:

May 31, 2021 at 2:21 pm

I am stuck with the hardware part could u please guide me ?

Reply



Bladeron says:

April 2, 2020 at 8:26 am

For the “not super electrical soldering skills guys” like me could you add a picture of the board with the cables? XD

Could project I'll try it soon!

Reply



AL says:

April 8, 2020 at 8:11 am

Hi dirk,

I am designing a lighting product and in need of a system designer to help develop a custom POV system.

I have the LED product and processor for regular video projection, if you can consult with me on modifying this system to POV?

Some products for reference:

<https://www.indiegogo.com/projects/magilight-creativity-without-boundaries#/>

<http://www.thepixelstick.com/>

<https://www.lightpaintingstick.com/>

Best,

Reply



Bram says:

May 14, 2020 at 9:16 pm

Hi Dirk,

Your code on the site is missing some libraries. are they not printed on the site because of the "less than" and "greater then" signs. Line 31 through 36.

The first two are probably Adafruit_NeoPixel.h and SD.h.

But what are the rest?

I'm guessing Adafruit_SSD1306.h for the display and EEPROM.h on line 66

Thanks for the help.

Reply



Tobias says:

November 23, 2020 at 6:26 am

Good Morning Bram,
could you solve the issue. I had the same when I try to compile.

Reply



Russ says:

May 22, 2020 at 9:36 am

Hi Dirk, thanks for sharing your awesome project. One question, in your code listing any #include statements are missing what's actually included! Would you be able to rectify that?

Reply



sanyam says:

June 19, 2021 at 8:56 am

```
#include // Library for the WS2812 Neopixel Strip
#include // Library for the SD Card
#include
#include
#include
#include
```

Reply



Flo says:

September 19, 2020 at 2:42 pm

Hello,

I have this project in progress, and after bought all of the recommended parts I am stuck at the coding part. At line 65 I am getting an error : sketch_sep19a:65:10: fatal error: FILENAME: No such file or directory

```
#include "FILENAME"
```

```
^~~~~~
```

compilation terminated.

exit status 1

FILENAME: No such file or directory

Can you help me please ?

Thanks in advance

Reply



Jose Alejandro says:

October 8, 2020 at 5:34 am

Hi Dirk,

Hope everything is going well. I was trying to compile the programme but it comes with an error in this line which I do not know what to include,

```
// EEPROM setup. saves all values in eeprom. the mega has 4kb of eeprom storage
```

```
#include ????
```

Could you be so kind and help me to figure out what to do? I am not a programmer and this is difficult for me.

Thank you

Reply

timy – the open source camera timer – Dirk Essl says:

October 23, 2020 at 5:35 pm

[...] WiFi or Bluetooth for later updates (remote trigger for lighty)

[...]



Reply

How I created my own DIY camera timer for \$50 - DIY Photography

says:

October 26, 2020 at 3:01 pm

[...] WiFi or Bluetooth for later updates (remote trigger for lighty)

[...]

Reply

How I created my own DIY camera timer for \$50 – Photography

says:

October 26, 2020 at 4:19 pm

[...] WiFi or Bluetooth for later updates (remote trigger for lighty)

[...]

Reply

***How I created my very own DIY digital camera timer for \$50 -
Bortz Tech*** says:

October 29, 2020 at 9:51 am

[...] WiFi or Bluetooth for later updates (distant set off for lighty)

[...]

Reply



Harry says:

December 1, 2020 at 11:20 pm

Though I haven't got all the parts yet (waiting for lights), I decided to test and compile and had the same errors some people have said.

After some googling, two things I did to get it to compile.

1 – added the Adafruit NeoPixel library

2 – Line 65 – #include (was missing).

Hopefully it will for compile for others too and can test with hardware. Feedback would be great.

Reply



sanyam says:

June 19, 2021 at 8:58 am

check his new code which he posted on his site its complete mine compiled

Reply



Harry says:

December 5, 2021 at 12:02 am

Hi sanyam, are you refering to the code on this blog page or elsewhere?

As I know, Dirk had started to use different components, instead of the mega2560 etc from a different site.

Unless he carried on with this stick too.

Reply



Yash Jariwala says:

March 20, 2021 at 5:46 am

My LIGHTHY shows sd card initializing fail...

please help me...

and you made mistake in diagram that CS pin should be connected with pin D53.

Reply



Achim says:

April 14, 2021 at 11:38 am

Hello Dirk,

first of all, I am totally enthusiastic about your project. Really good work.

I have rebuilt everything as you described it. The sketch compiles perfectly and I have checked the position of the cables several times. But I get an error at startup.


SD init failed!

Do you have an idea what's the reason for?

Best regards

Achim

Reply

 **Cómo construir un 'Pixelstick' de bricolaje por aproximadamente \$ 22** says:

May 9, 2021 at 1:13 pm

[...] Puedes ver más de su trabajo en su sitio web, o siguiéndolo en Facebook y YouTube. Este artículo también fue publicado aquí. [...]

Reply



Thomas Klee says:

May 20, 2021 at 7:22 pm

Hello Dirk, i have build this device, but i become show the message" SD init failed" Can you help me.

THX

Thomas from Germany, Black Forest

Reply



Thomas Klee says:

May 21, 2021 at 6:06 am

Hallo Dirk,

habe deinen Stick 1 zu 1 mit den aufgeführten Komponenten nachgebaut. Sobald 5V angelegt werden erscheint auf dem Display der Starttext und am Ende SD init failed und das Display geht aus. Kannst du mir hier weiterhelfen, was könnte der Fehler sein.

Viele Grüße aus dem Schwarzwald

Thomas

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

☐

Save my name, email, and website in this browser for the next time I comment.

[Post Comment](#)

ELSEWHERE

DIRK ESSL

Creator of www.caffenol.org, Lightroom LensTagger and the PolaDock.
Coffee Addict, Halftime Nerd



CALM NIGHTS



SPONSORS

© Dirk Essl 2022