

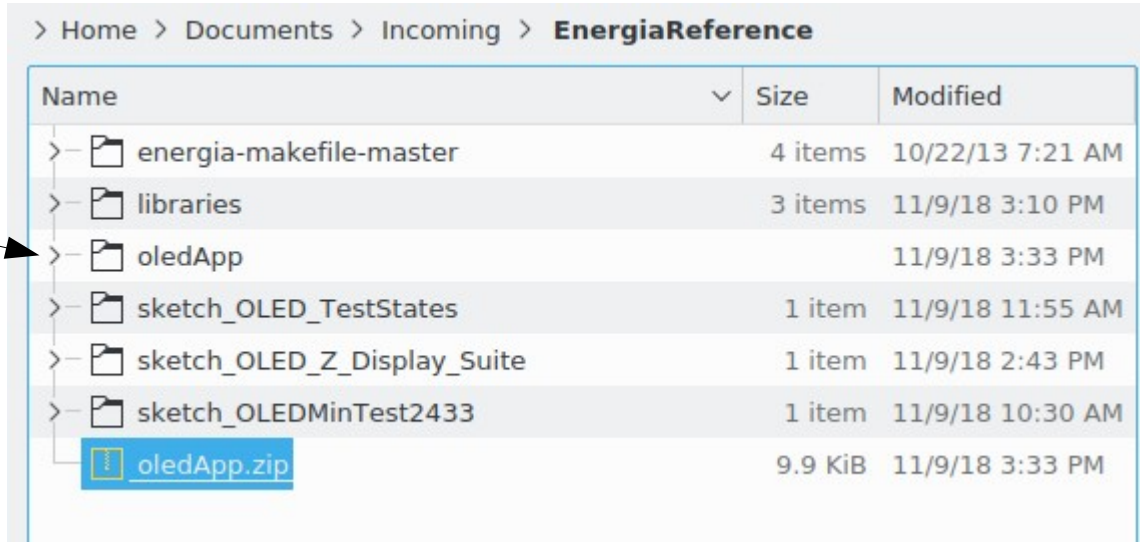
M14-Exercise 4 for M14-Assignment

This demo uses the Wire library I2C OLED display and Launchpad Board in oledApp.ino

This dynamically displays alternating text and bitmap patterns

Download oledApp.zip from the web site
Place in the Energia folder (not libraries)

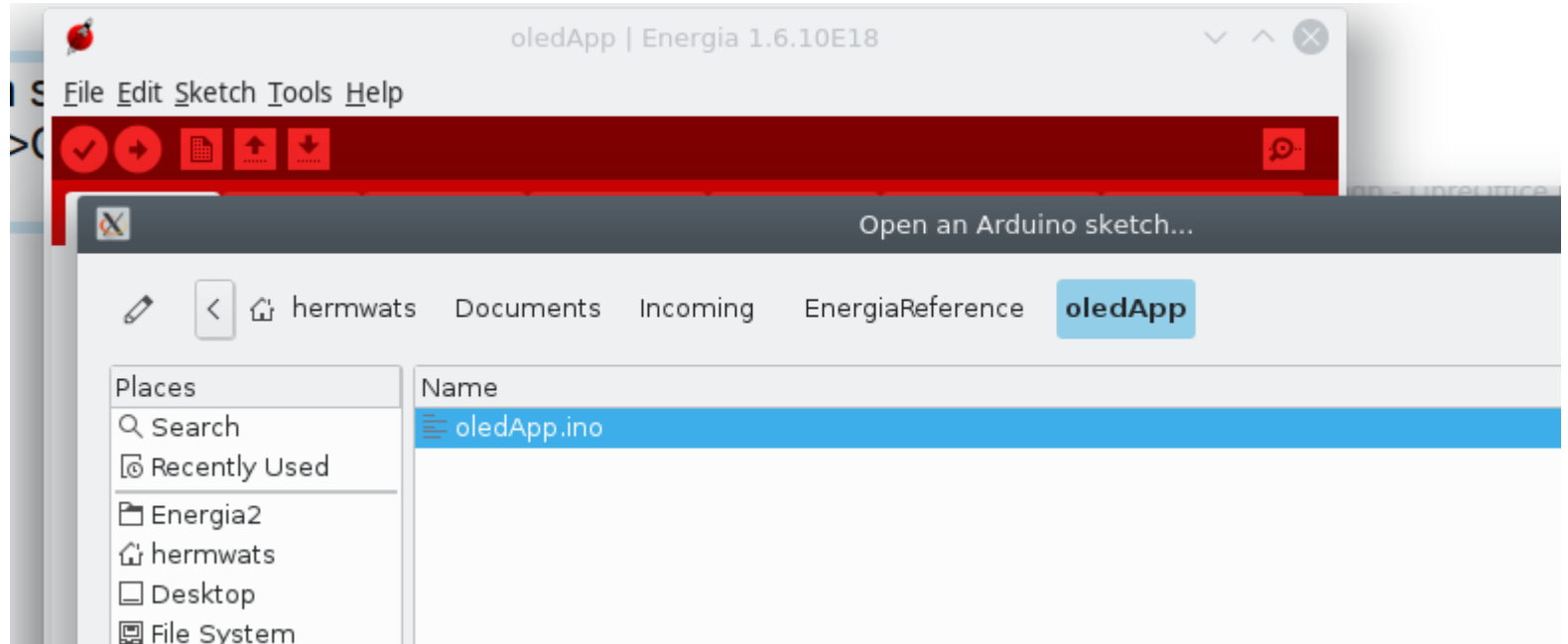
UNZIP to create an oledApp folder



> Home > Documents > Incoming > **EnergiaReference**

Name	Size	Modified
> - [Folder] energia-makefile-master	4 items	10/22/13 7:21 AM
> - [Folder] libraries	3 items	11/9/18 3:10 PM
> - [Folder] oledApp		11/9/18 3:33 PM
> - [Folder] sketch_OLED_TestStates	1 item	11/9/18 11:55 AM
> - [Folder] sketch_OLED_Z_Display_Suite	1 item	11/9/18 2:43 PM
> - [Folder] sketch_OLEDMinTest2433	1 item	11/9/18 10:30 AM
[File] oledApp.zip	9.9 KiB	11/9/18 3:33 PM

Then select
File->Open->oledApp->oledApp.ino





oledApp

Font.h

MyTest.h

MyTest2.h

images.h

imagesBak.h

testimage. ▾

```
1 //
2 // http://www.electronicwings.com/ti-launchpad/
3 // oled-graphic-display-interfacing-with-msp-exp430g2-ti-launchpad
4 // 20180506 - HW
5
6 #include <Wire.h>
7 #include "Font.h"
8 #include <string.h>
9 #include "imagesBak.h"
10
11 #define OLED_Write_Address 0x3C
12
13 void OLED_Data(char *DATA) /* Function for sending data to OLED */
14 {
15     int len = strlen(DATA);
16     for (int g=0; g<len; g++)
17     {
18         for (int index=0; index<5; index++)
19         {
20             Wire.beginTransmission(OLED_Write_Address); /* Begin transmission to slave devi
21             /* Queue data to be transmitted */
```

Make sure all the tabs show up as shown here

File Edit Sketch Tools Help

oledApp

Font.h

MyTest.h

MyTest2.h

images.h

imagesBak.h

testimage.▼

```
1 //
2 // http://www.electronicwings.com/ti-launchpad/
3 // oled-graphic-display-interfacing-with-msp-exp430g2-ti-launchpad
4 // 20180506 - HW
5
6 #include <Wire.h>
7 #include "Font.h"
8 #include <string.h>
9 #include "imagesBak.h"
10
11 #define OLED_Write_Address 0x3C
12
13 void OLED_Data(char *DATA) /* Function for sending data to OLED */
14 {
15     int len = strlen(DATA);
16     for (int g=0; g<len; g++)
17     {
18         for (int index=0; index<5; index++)
19         {
20             Wire.beginTransmission(OLED_Write_Address); /* Begin transmission to slave device */
21             /* Queue data to be transmitted */
```

Compile and upload the demo

Done compiling.

```
Sketch uses 9,346 bytes (60%) of program storage space. Maximum is 15,360 bytes.
Global variables use 134 bytes (3%) of dynamic memory, leaving 3,962 bytes for local variables.
```

- Sketch_oled_test uses 9,346 bytes (60%) of program storage space.

Maximum is 15,360 bytes for MSP430FR2433.

Global variables use 134 bytes (3%) of dynamic memory, leaving 3,962 bytes for local variables. Maximum RAM is 4,096 bytes

- For Arduino Uno, example for Monochrome OLEDs based on SSD1306 drivers
Pick one up today in the adafruit shop! -----> http://www.adafruit.com/category/63_98

This example is for a 128x64 size OLED display using I2C to communicate

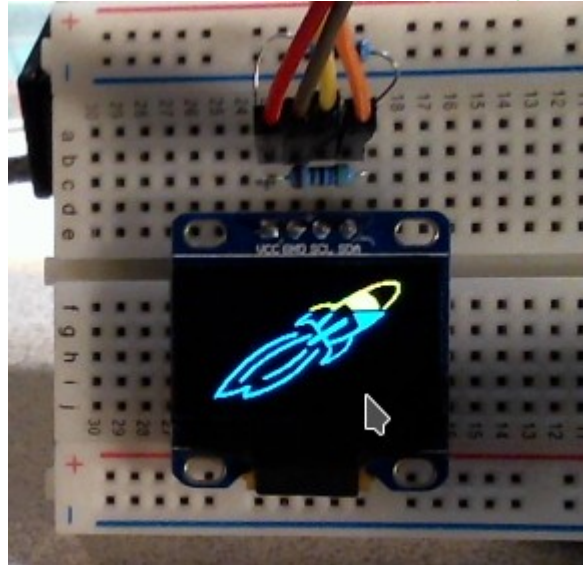
Sketch uses 14282 bytes (44%) of program storage space. Arduino Maximum is 32256 bytes. Global variables use 1501 bytes (73%) of dynamic memory, leaving 547 bytes for local variables.

Maximum is 2048 bytes.

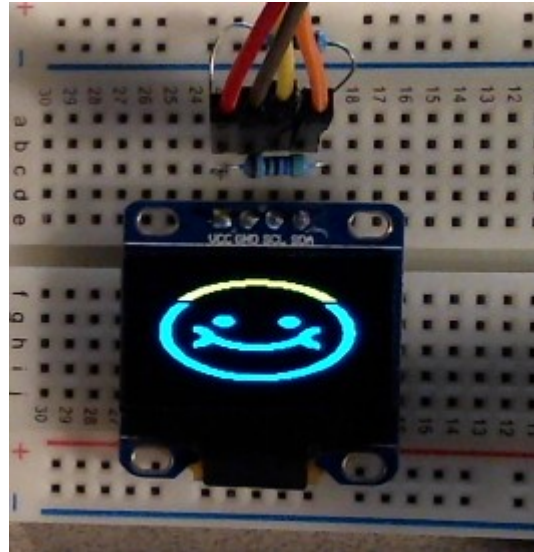
So even with a small Sketch_oled_test graphics application the amount of memory used is large.

- The Adafruit library uses the total memory capacity of the MSP430FR2433. Therefore the Adafruit library and derivatives is too big to run on this version of the MSP430. There are larger versions, but not used for this course.

```
118 void setup() {  
119     Wire.begin(); /* Initiate wire library and join I2C bus as a master */  
120     OLED_init(); /* Initialize OLED */  
121     delay(100);  
122     OLED_clear(); /* Clear OLED */  
123     delay(1000);  
124     OLED_image(Launchpad_Logo);  
125     delay(5000);  
126     OLED_clear();  
127     delay(200);  
128     OLED_setXY(0x31, 0x7F, 0x03, 0x02);  
129     OLED_Data("Smiley");  
130     OLED_setXY(0x36, 0x7F, 0x05, 0x05);  
131     OLED_Data("Demo");  
132     OLED_setXY(0x00, 0x7F, 0x00, 0x07);  
133     delay(3000);  
134 }
```



```
136 void loop() {  
137     OLED_image(Smileys_1);  
138     delay(500);  
139     OLED_image(Smileys_2);  
140     delay(500);  
141     OLED_image(Smileys_3);  
142     delay(500);  
143     OLED_image(Smileys_4);  
144     delay(1000);  
145 }
```



Diagonal line

```
146 byte MX, MY, PY, BY;
147 char OY;
148 // Diagonal line - bit mapping left to right sweep
149 // https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf
150 for(byte i=0; i<0x80; i++)
151 {
152     MX = i;           // Column (segment) number 0-127
153     MY = i>>1;       // 0-63 Y value row number
154     PY = MY>>3;      // Page # (Byte Row)
155     BY = MY&0x07;    // bit number (in Byte)
156     OY = 0x01<<BY;  // set bit
157
158     OLED_setXY(MX, 0x7F, PY, PY);
159     OLED_Binary(OY); // write Column byte i page
160     delay(100);
161 }
162 delay(2000);
163 }
```

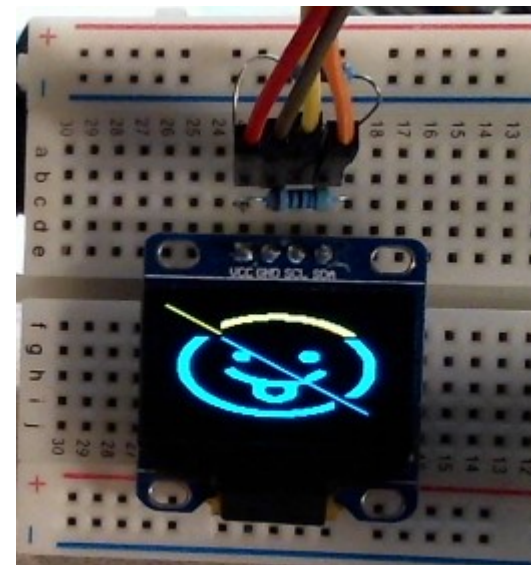


Figure 10-1 : Address Pointer Movement of Page addressing mode

	COL0	COL 1	COL 126	COL 127
PAGE0	→				
PAGE1	→				
:	:	:	:	:	:
PAGE6	→				
PAGE7	→				