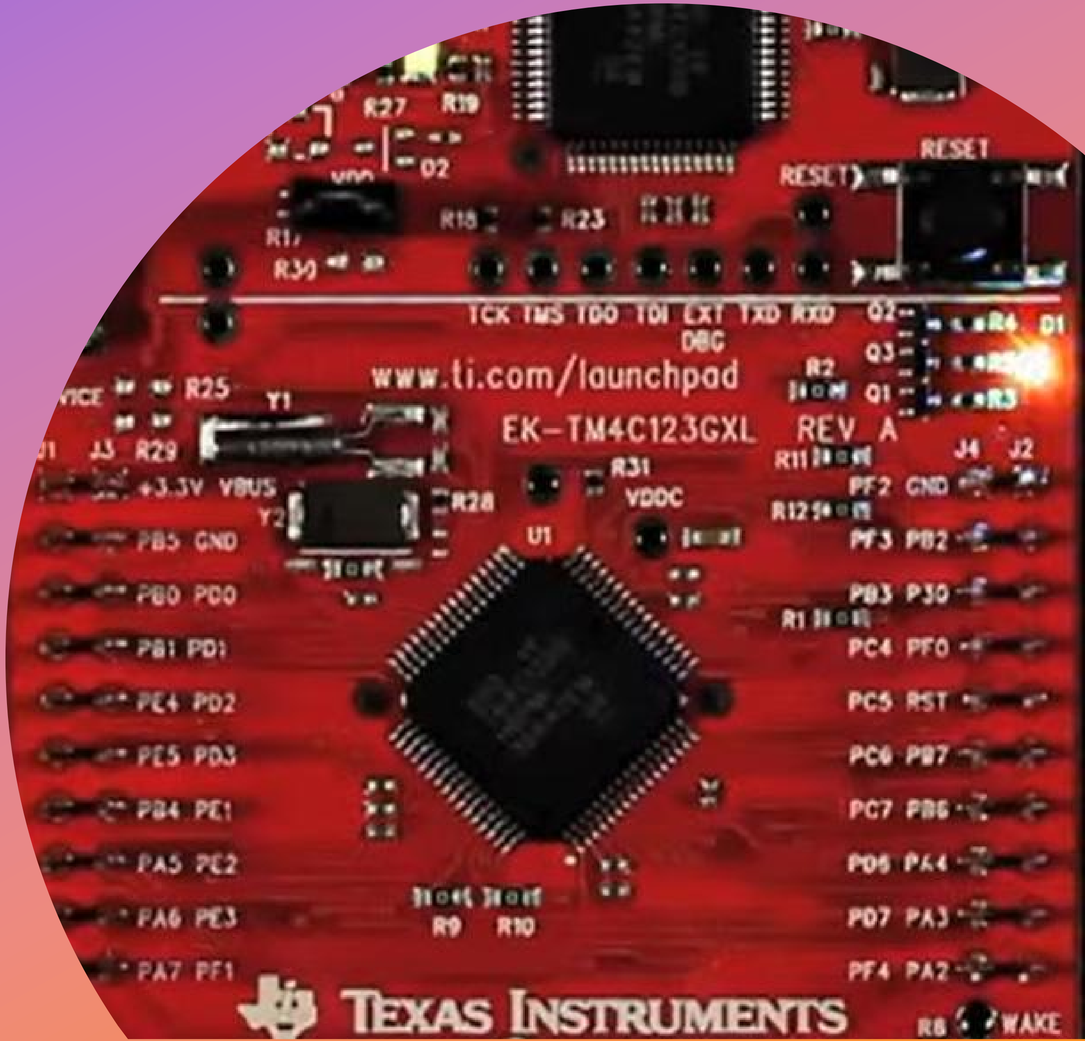
ELECTRONIC & SIGNAL

Guo Xiaofan

Ketul

Liu Yang

Umut



Introduction

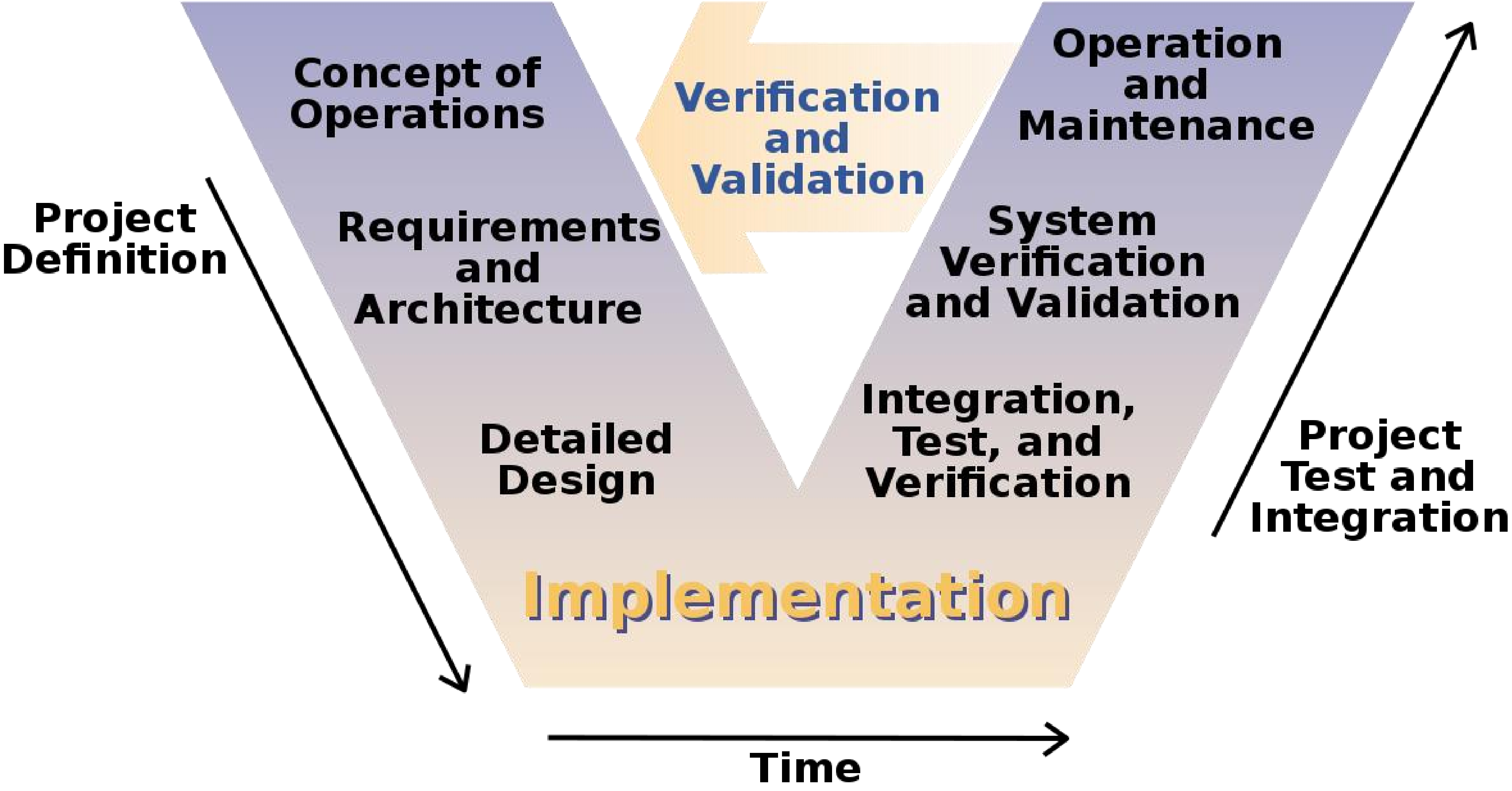
Mission 1:

Microcontroller programming discovery

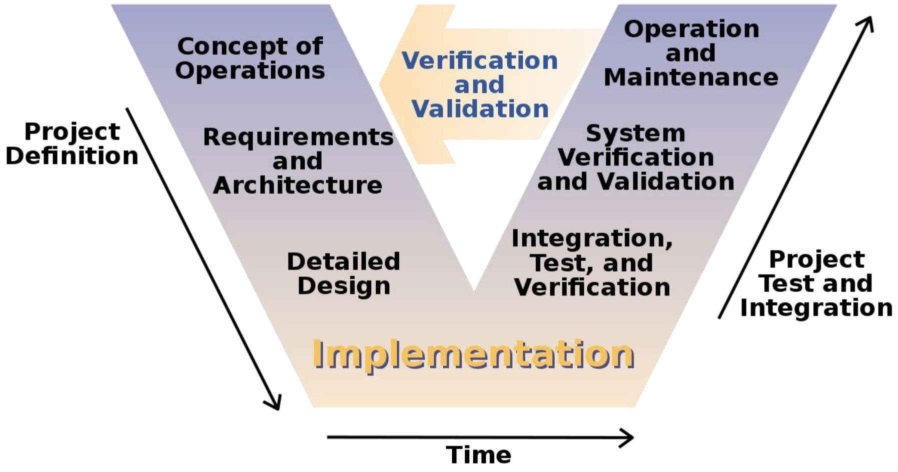
1 - MISSION

1) V - Mode

V Mode


Concept of Operations
Requirements and Architecture
Detailed Design
Verification and Validation
Operation and Maintenance
System Verification and Validation
Integration, Test, and Verification
Implementation
Project Definition
Project Test and Integration
Time

V Mode


Concept of Operations
Verification and Validation
Operation and Maintenance
Project Definition
Requirements and Architecture
System Verification and Validation
Detailed Design
Integration, Test, and Verification
Project Test and Integration
Implementation
Time

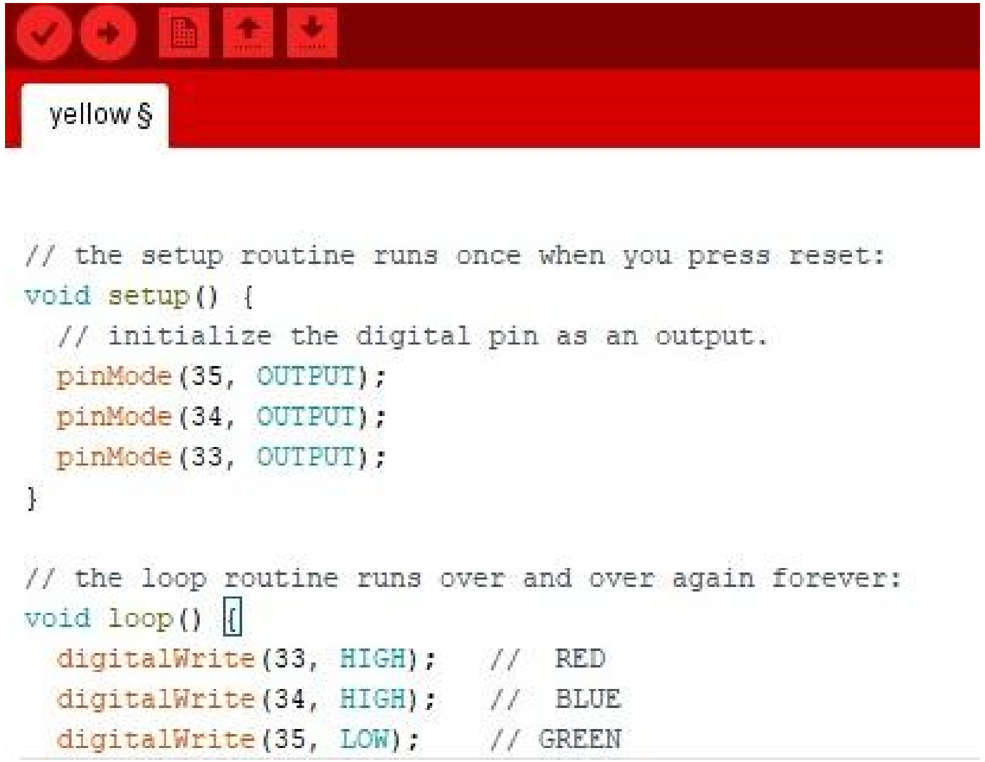
The V-cycle, often associated with software development, is a structured model that divides the development process into phases shaped like a "V." It begins with requirements gathering and high-level planning, proceeds through design and coding, and then ascends with various testing stages, culminating in validation and deployment. The key concept is rigorous validation and testing at each stage to ensure the final product aligns with initial requirements. This approach is known for its thoroughness and suitability for projects with well-defined requirements and minimal expected changes

1 - MISSION

2) Tricolor LED:

BLUE – RED – GREEN – YELLOW

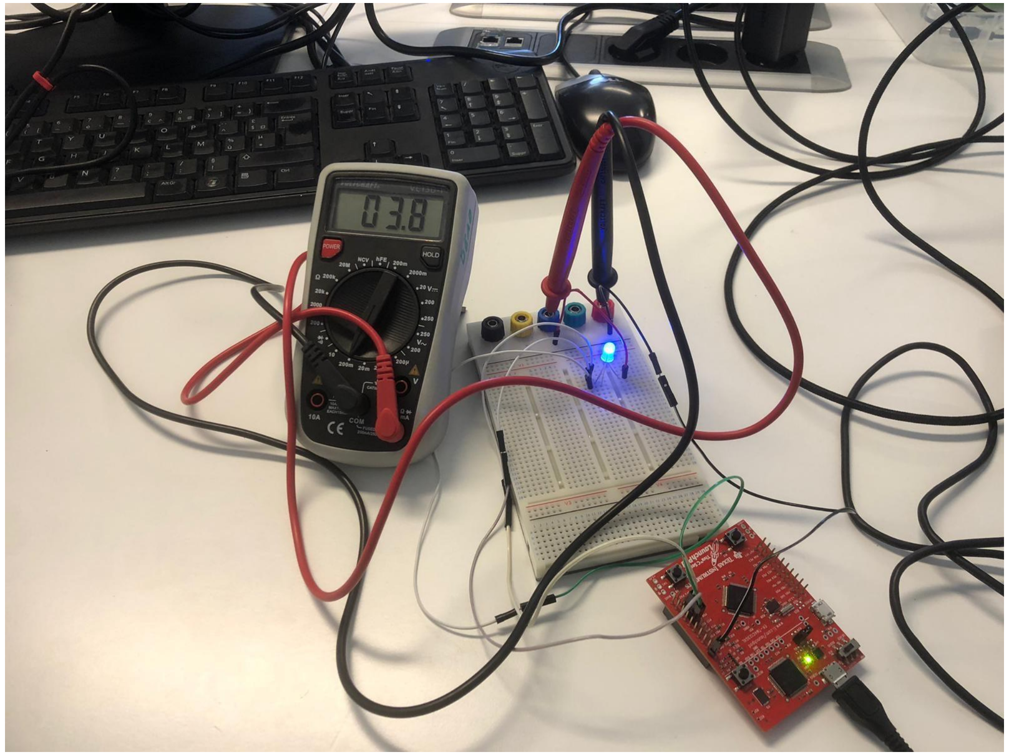
Blue – Red – Green – Yellow



```
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(35, OUTPUT);
  pinMode(34, OUTPUT);
  pinMode(33, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(33, HIGH);   //  RED
  digitalWrite(34, HIGH);   //  BLUE
  digitalWrite(35, LOW);    // GREEN
```

Modify based on the example code to control the output voltage of different interfaces to display LEDs in different colors, as shown below:

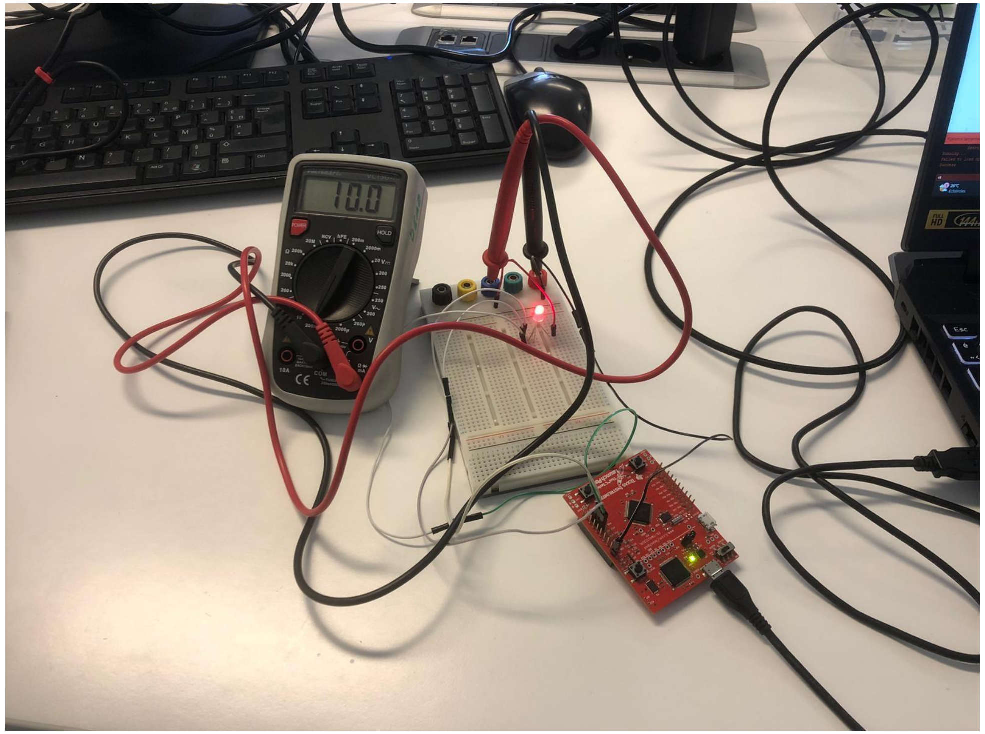
Blue – Red – Green – Yellow



Blue LED (3.8mA):

Blue LEDs can produce a significant amount of light even at lower currents.

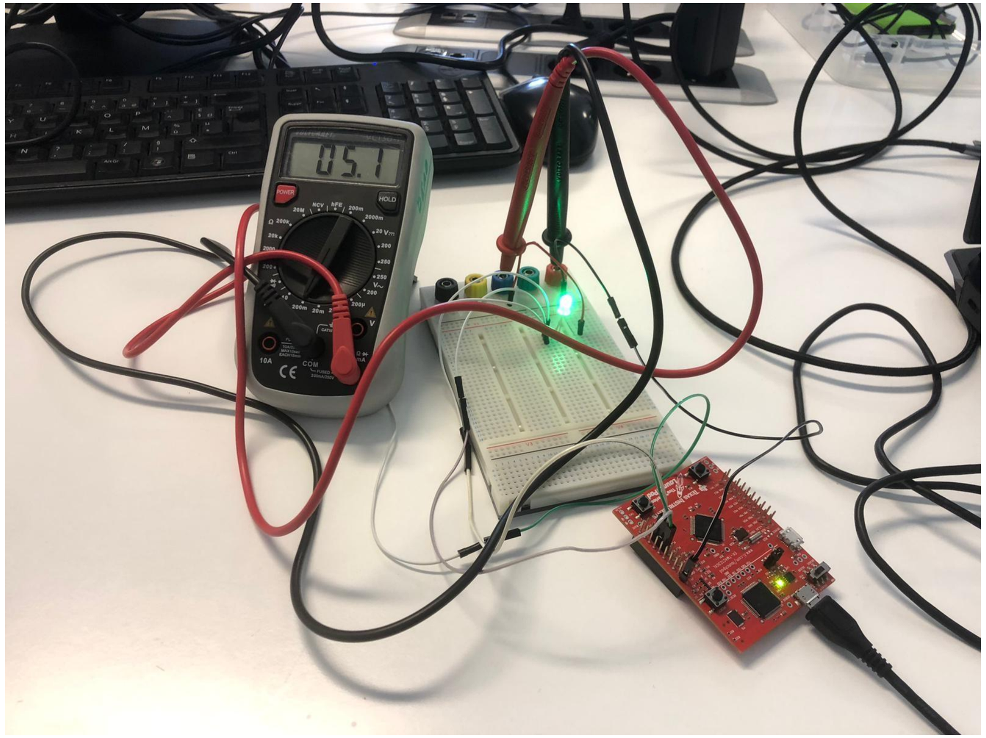
Blue – Red – Green – Yellow



Red LED (10.0mA):

To produce a visible amount of light, red LEDs require a higher forward current of 10.0mA due to the higher forward voltage drop.

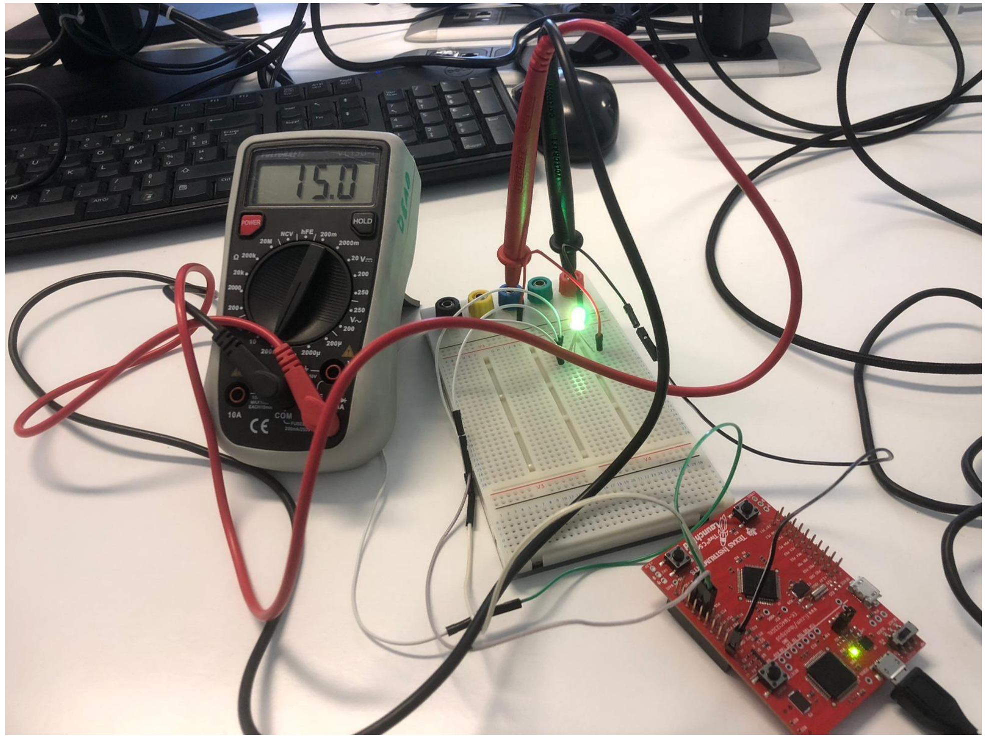
Blue – Red – **Green** – Yellow



Green LED (5.1mA):

The lower forward current drop allows green LEDs to achieve the desired luminosity at a lower forward current.

Blue – Red – Green – Yellow

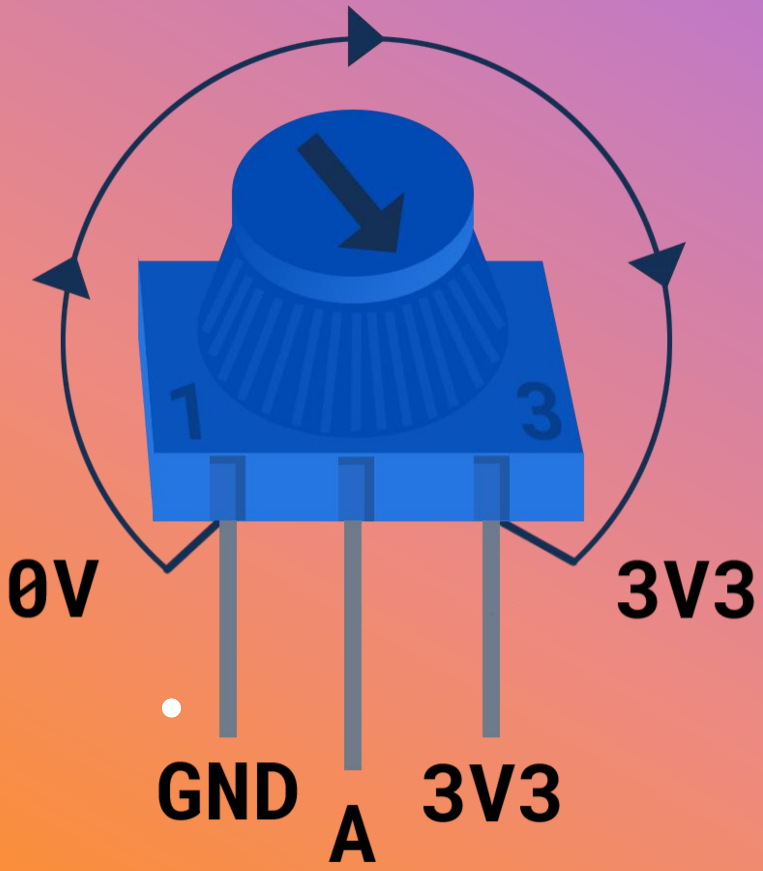


Yellow LED (15.0mA):

We use the red + green method to mix yellow, so the yellow excitation current is equal to the sum of the red and green currents (within the error range).

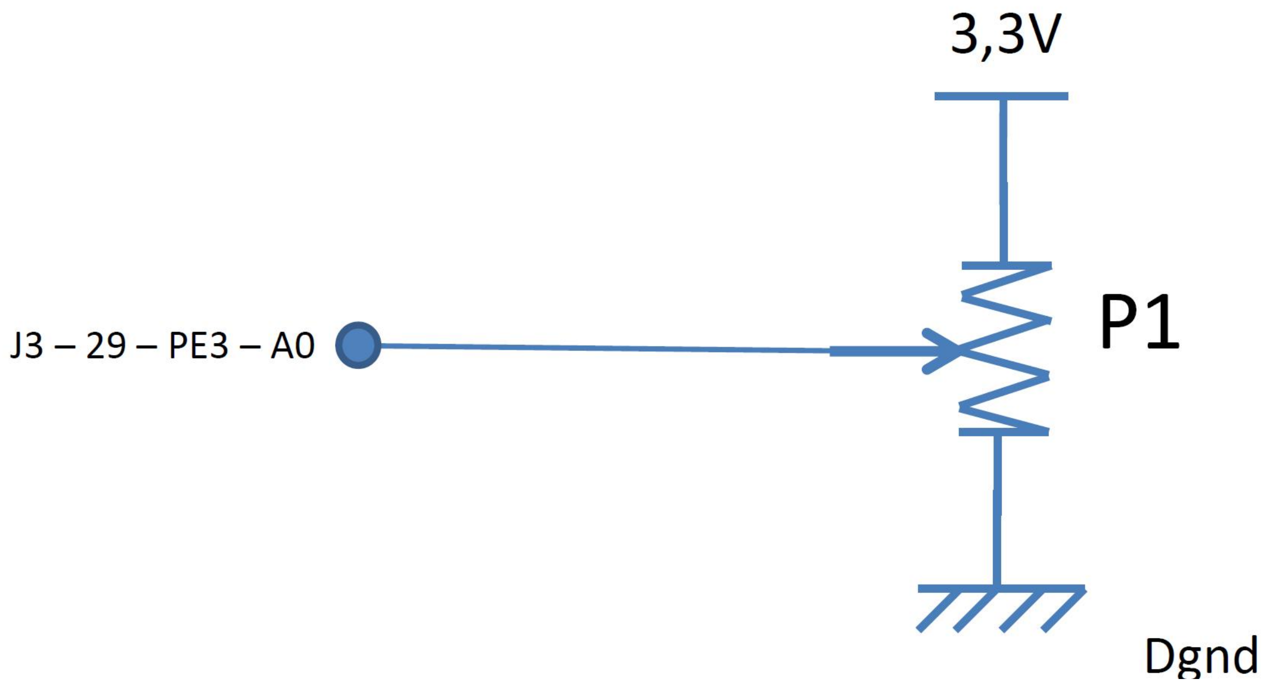
1 - MISSION

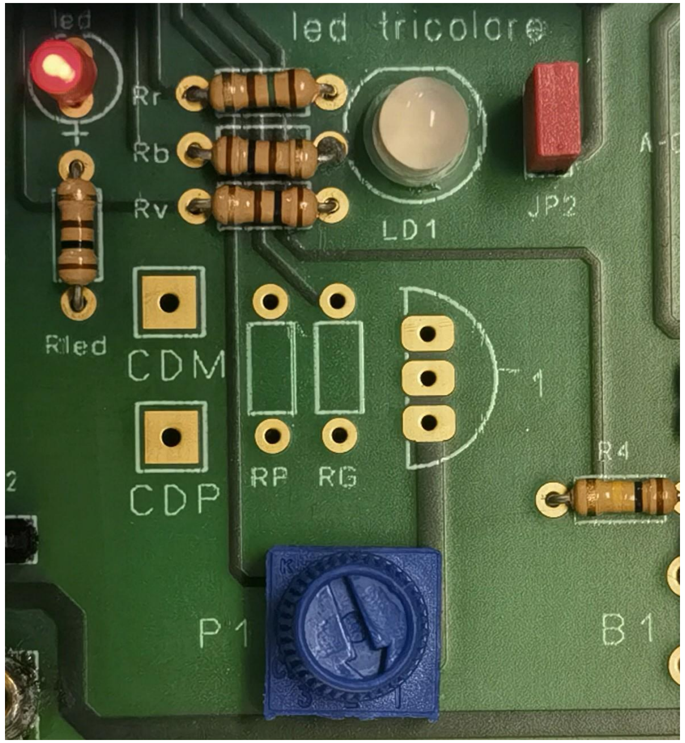
3) Connect a Potentiometer and Read the Analog Value



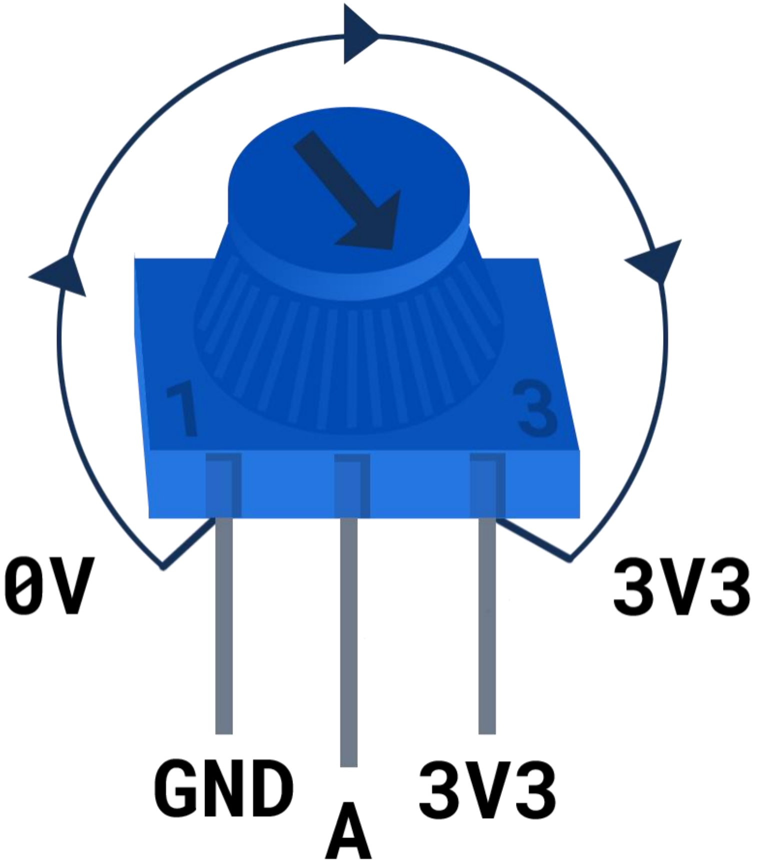
Potentiometer

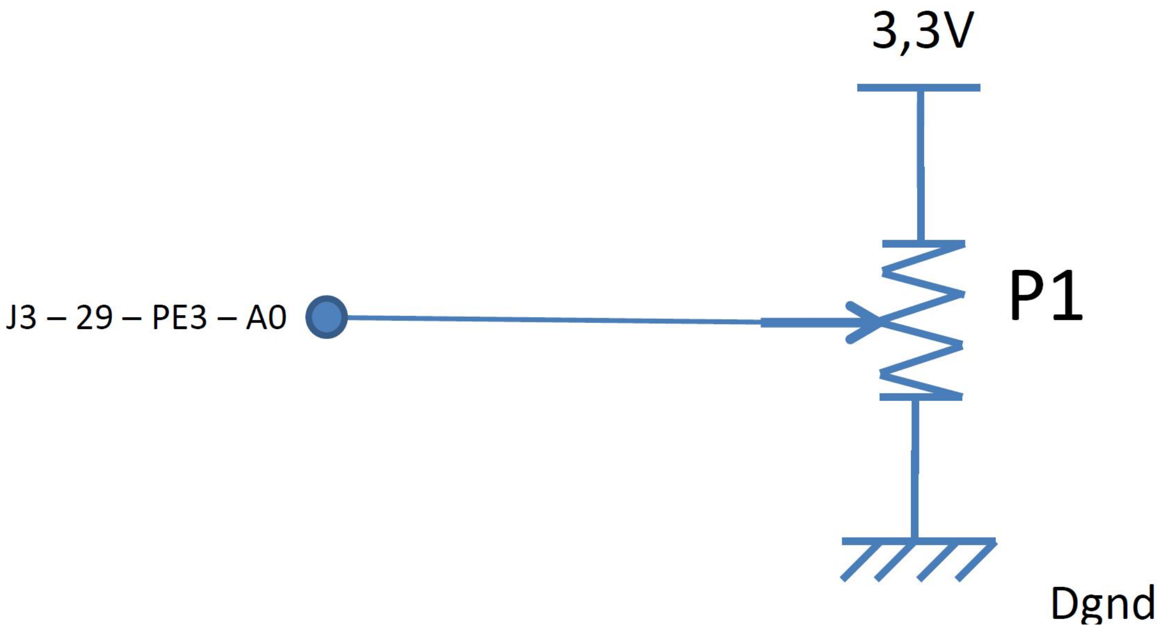
Parameter	Parameter Name	Nom	Unit
V_{DD}	VDD Supply Voltage	3.3	Voltage



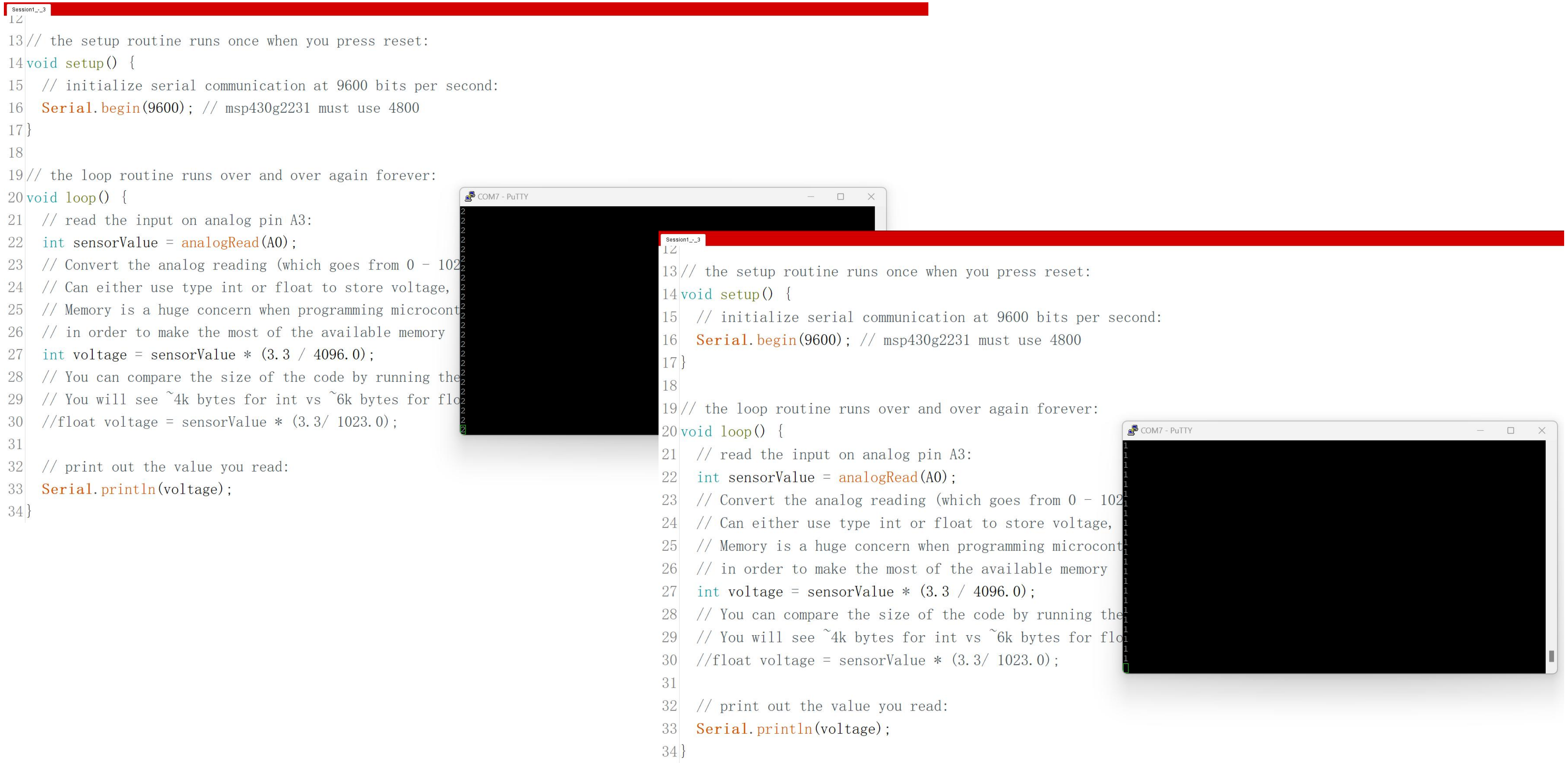


Content





Content



```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600); // msp430g2231 must use 4800
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin A3:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 102
  // Can either use type int or float to store voltage,
  // Memory is a huge concern when programming microcont
  // in order to make the most of the available memory
  int voltage = sensorValue * (3.3 / 4096.0);
  // You can compare the size of the code by running the
  // You will see ~4k bytes for int vs ~6k bytes for flo
  //float voltage = sensorValue * (3.3/ 1023.0);

  // print out the value you read:
  Serial.println(voltage);
}
```

```
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
2
```

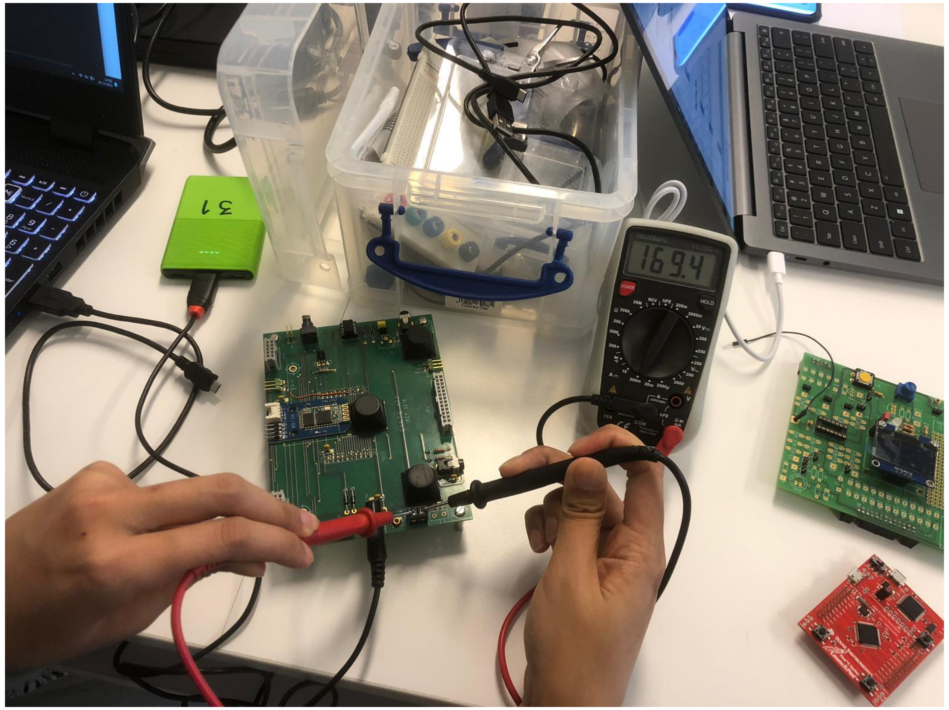
```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600); // msp430g2231 must use 4800
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin A3:
  int sensorValue = analogRead(A0);
  // Convert the analog reading (which goes from 0 - 102
  // Can either use type int or float to store voltage,
  // Memory is a huge concern when programming microcont
  // in order to make the most of the available memory
  int voltage = sensorValue * (3.3 / 4096.0);
  // You can compare the size of the code by running the
  // You will see ~4k bytes for int vs ~6k bytes for flo
  //float voltage = sensorValue * (3.3/ 1023.0);

  // print out the value you read:
  Serial.println(voltage);
}
```

```
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
1
```

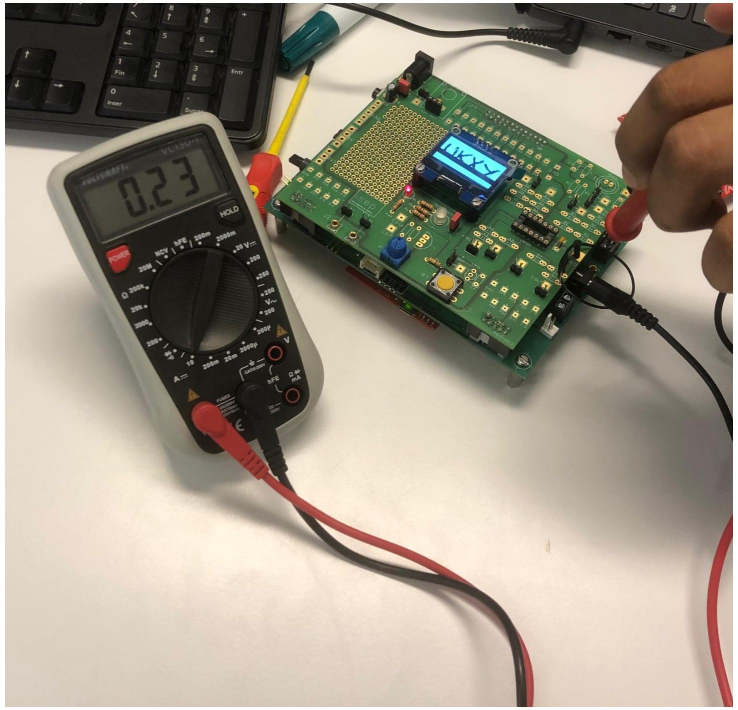
1 - MISSION

4) Measure the Power Consumption of the Board

Power Consumption



Current when Bluetooth Active

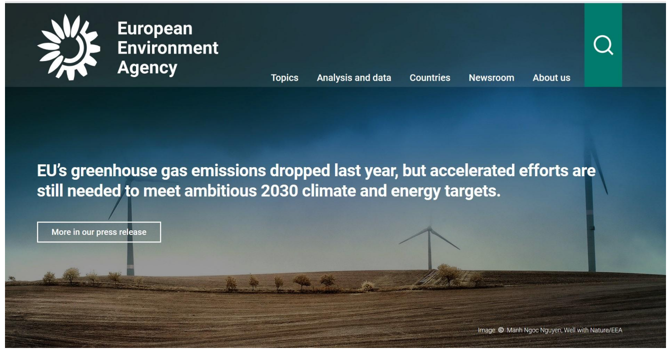


Current when Bluetooth & OLED Active

Power Consumption

Condition	Power Bank Ootput (V)	Currnet (A)	Power (W)	Rate	CO$_2$ (G)
Bluetooth	5	0.1694	0.847	0.059	0.049973
Bluetooth & OLED	5	0.23	1.15		0.06785

RenSmart. “KWH-To- CO2.” Rensmart.com, 2016, www.rensmart.com/Calculators/KWH-to-CO2.



EU's greenhouse gas emissions dropped last year, but accelerated efforts are still needed to meet ambitious 2030 climate and energy targets.

Image © Manh Ngoc Nguyen, Well with Nature/EEA

1 - MISSION

5) The Microcontroller Parameters

TEXAS INSTRUMENTS-PRODUCTION DATA

TEXAS INSTRUMENTS

Tiva™ TM4C123GH6PM Microcontroller

DATA SHEET

DS-TM4C123GH6PM-15842.2741
SPMS376E

Copyright © 2007-2014
Texas Instruments Incorporated

Memories

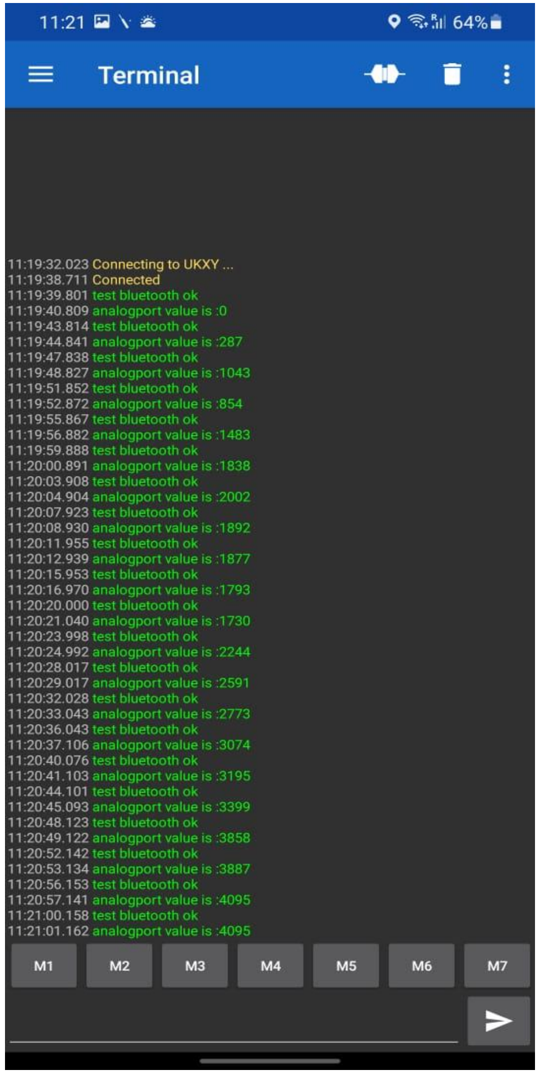
Inter Memory		
Tyep	Size	Functino
SRAM	32 KB	
ROM		
Flash Memory	256KB	
EEPROM	2KB	

1 - MISSION

6) Access to the Bluetooth link

Bluetooth link

```
int analogport =29;
int value =0;

void setupO {
pinMode ( analogport , INPUT );
Serial . begin (9600);
Serial1.begin(9600);
Serial1.read();

void loopO {
 // put your main code here , to run repeatedly :
 Serial . printin (' test serial ok");
 Serial1.printIn(" test bluetooth ok");
 delay (1000);
 value = analogead ( analogport);
 Serial1.print("analogport value is :");
 Serial1.print(value);
 Serial1.println();
 delay (2000);
```

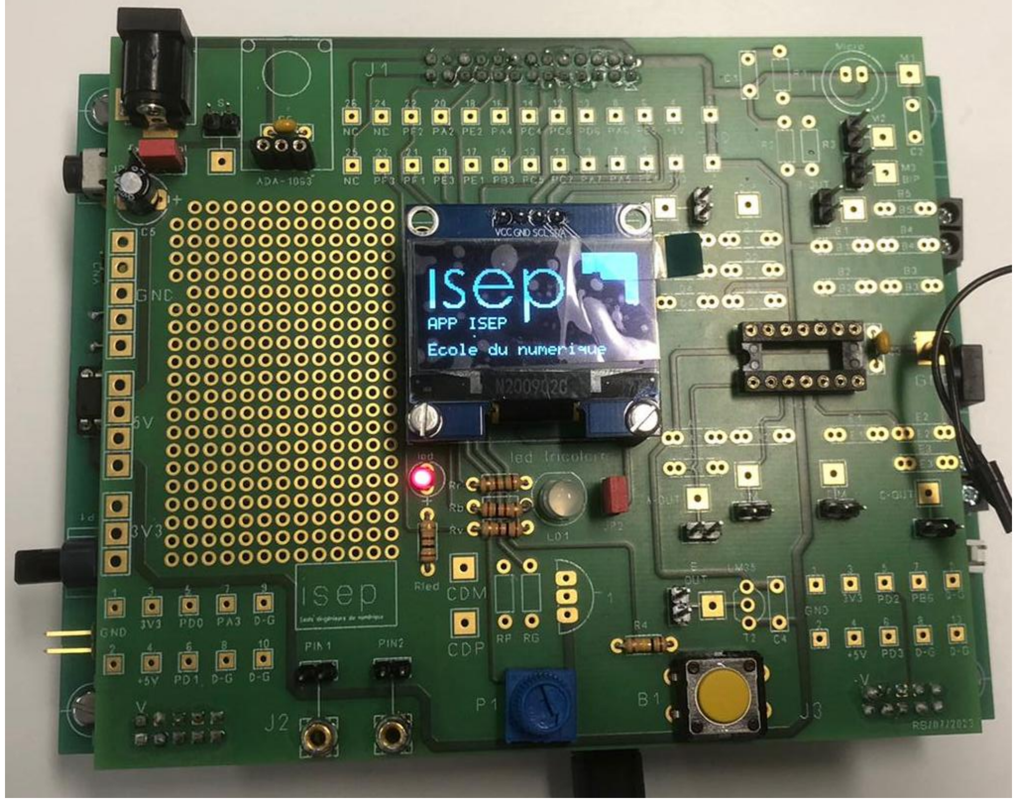


```
11:19:32.023 Connecting to UKXY ...
11:19:38.711 Connected
11:19:39.801 test bluetooth ok
11:19:40.809 analogport value is :0
11:19:43.814 test bluetooth ok
11:19:44.841 analogport value is :287
11:19:47.838 test bluetooth ok
11:19:48.827 analogport value is :1043
11:19:51.852 test bluetooth ok
11:19:52.872 analogport value is :854
11:19:55.867 test bluetooth ok
11:19:56.882 analogport value is :1483
11:19:59.888 test bluetooth ok
11:20:00.891 analogport value is :1838
11:20:03.908 test bluetooth ok
11:20:04.904 analogport value is :2002
11:20:07.923 test bluetooth ok
11:20:08.930 analogport value is :1892
11:20:11.955 test bluetooth ok
11:20:12.939 analogport value is :1877
11:20:15.953 test bluetooth ok
11:20:16.970 analogport value is :1793
11:20:20.000 test bluetooth ok
11:20:21.040 analogport value is :1730
11:20:23.998 test bluetooth ok
11:20:24.992 analogport value is :2244
11:20:28.017 test bluetooth ok
11:20:29.017 analogport value is :2591
11:20:32.028 test bluetooth ok
11:20:33.043 analogport value is :2773
11:20:36.043 test bluetooth ok
11:20:37.106 analogport value is :3074
11:20:40.076 test bluetooth ok
11:20:41.103 analogport value is :3195
11:20:44.101 test bluetooth ok
11:20:45.093 analogport value is :3399
11:20:48.123 test bluetooth ok
11:20:49.122 analogport value is :3858
11:20:52.142 test bluetooth ok
11:20:53.134 analogport value is :3887
11:20:56.153 test bluetooth ok
11:20:57.141 analogport value is :4095
11:21:00.158 test bluetooth ok
11:21:01.162 analogport value is :4095
```

1 - MISSION

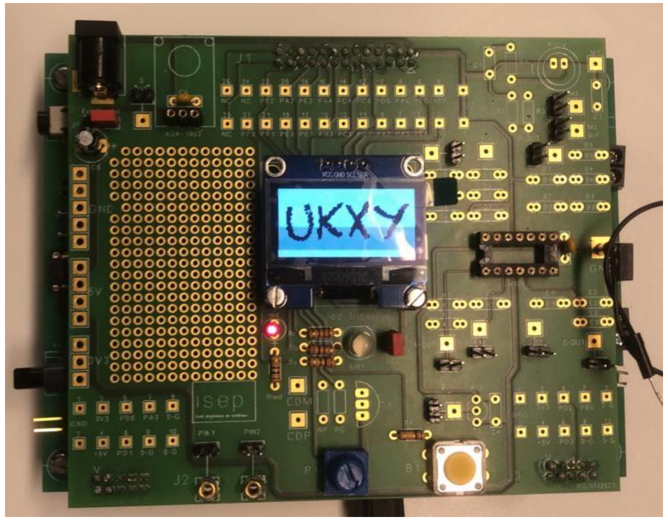
7) Sold the OLED display

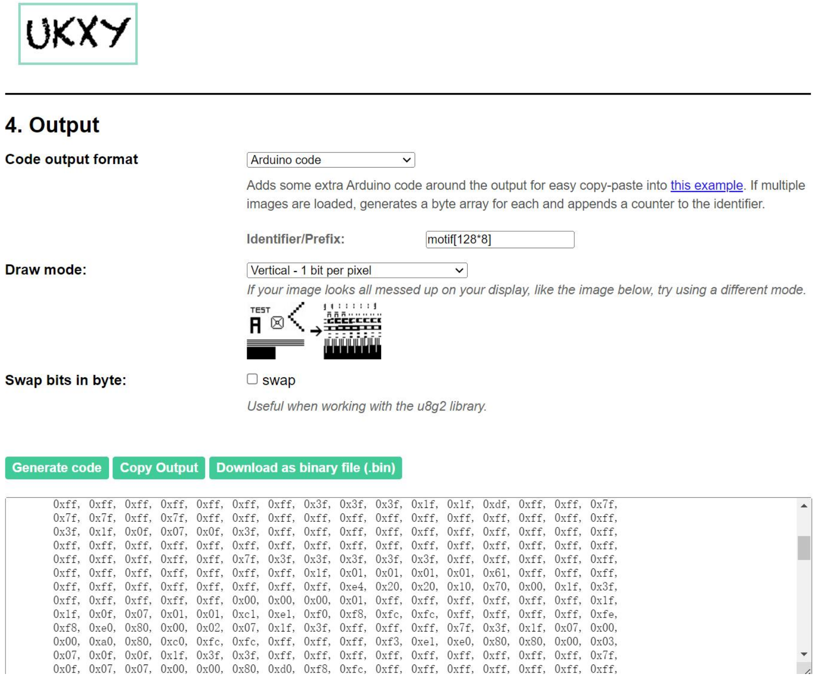
OLDE



Load given library to display messages.

OLDE





4. Output

Code output format Arduino code

Adds some extra Arduino code around the output for easy copy-paste into this example. If multiple images are loaded, generates a byte array for each and appends a counter to the identifier.

Identifier/Prefix: motif[128*8]

Draw mode: Vertical - 1 bit per pixel

If your image looks all messed up on your display, like the image below, try using a different mode.

Swap bits in byte: □ swap

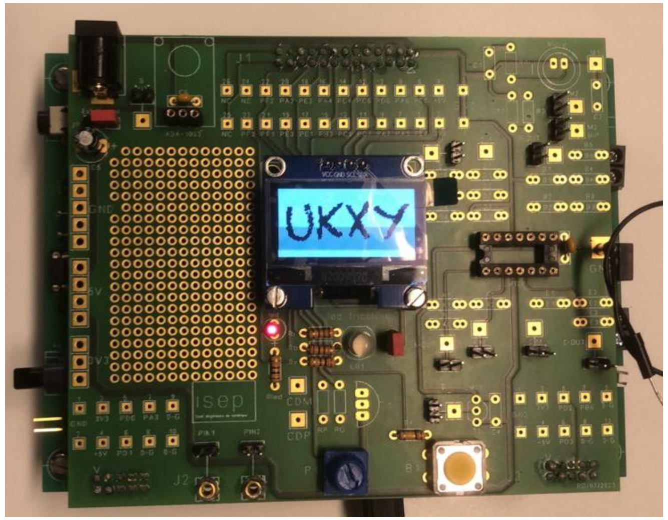
Useful when working with the u8g2 library.

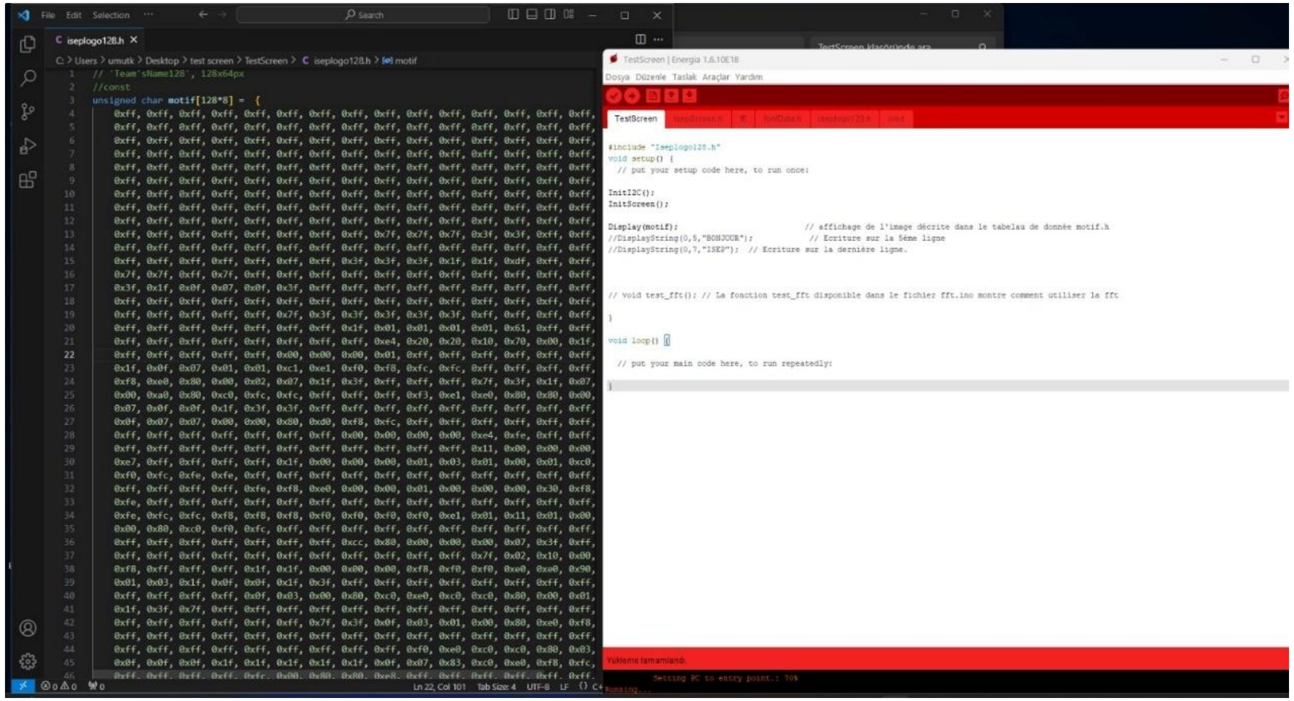
```
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0x3f, 0x3f, 0x1f, 0x1f, 0xdf, 0xff, 0xff, 0x7f,
0x7f, 0x7f, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0x3f, 0x1f, 0x0f, 0x07, 0x0f, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0x3f, 0x3f, 0x3f, 0x3f, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x1f, 0x01, 0x01, 0x01, 0x01, 0x61, 0xff, 0xff, 0xff,
0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe4, 0x20, 0x20, 0x10, 0x70, 0x00, 0x1f, 0x3f,
0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x1f,
0x1f, 0x0f, 0x07, 0x01, 0x01, 0xc1, 0xe1, 0xf0, 0xf8, 0xfc, 0xfc, 0xff, 0xff, 0xff, 0xff, 0xfe,
0xf8, 0xe0, 0x80, 0x00, 0x02, 0x07, 0x1f, 0x3f, 0xff, 0xff, 0xff, 0x7f, 0x3f, 0x1f, 0x07, 0x00,
0x00, 0xa0, 0x80, 0xc0, 0xfc, 0xfc, 0xff, 0xff, 0xff, 0xf3, 0xe1, 0xe0, 0x80, 0x80, 0x00, 0x03,
0x07, 0x0f, 0x0f, 0x1f, 0x3f, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7f,
0x0f, 0x07, 0x07, 0x00, 0x00, 0x80, 0xd0, 0xf8, 0xfc, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
```

1st

Convert the team LOGO icon to 128*64 size, and then converted to encoding

OLDE




File Edit Selection ···
Search
iseplogo128.h
C: > Users > umutk > Desktop > test screen > TestScreen > iseplogo128.h > motif

```
// 'Team'sName128', 128x64px
//const
unsigned char motif[128*8] = {
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0x7f, 0x7f, 0x3f, 0x3f, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x3f, 0x3f, 0x3f, 0x1f, 0x1f, 0xdf, 0xff, 0xff,
    0x7f, 0x7f, 0xff, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0x3f, 0x1f, 0x0f, 0x07, 0x0f, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0x3f, 0x3f, 0x3f, 0x3f, 0x3f, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x1f, 0x01, 0x01, 0x01, 0x01, 0x61, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xe4, 0x20, 0x20, 0x10, 0x70, 0x00, 0x1f,
    0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x01, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0x1f, 0x0f, 0x07, 0x01, 0x01, 0xc1, 0xe1, 0xf0, 0xf8, 0xfc, 0xfc, 0xff, 0xff, 0xff, 0xff,
    0xf8, 0xe0, 0x80, 0x00, 0x02, 0x07, 0x1f, 0x3f, 0xff, 0xff, 0xff, 0x7f, 0x3f, 0x1f, 0x07,
    0x00, 0xa0, 0x80, 0xc0, 0xfc, 0xfc, 0xff, 0xff, 0xff, 0xf3, 0xe1, 0xe0, 0x80, 0x80, 0x00,
    0x07, 0x0f, 0x0f, 0x1f, 0x3f, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0x0f, 0x07, 0x07, 0x00, 0x00, 0x80, 0xd0, 0xf8, 0xfc, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x00, 0x00, 0x00, 0x00, 0xe4, 0xfe, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x11, 0x00, 0x00, 0x00,
    0xe7, 0xff, 0xff, 0xff, 0xff, 0x1f, 0x00, 0x00, 0x00, 0x01, 0x03, 0x01, 0x00, 0x01, 0xc0,
    0xf0, 0xfc, 0xfe, 0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xfe, 0xf8, 0xe0, 0x00, 0x00, 0x01, 0x00, 0x00, 0x00, 0x30, 0xf8,
    0xfe, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xfe, 0xfc, 0xfc, 0xf8, 0xf8, 0xf8, 0xf0, 0xf0, 0xf0, 0xf0, 0xe1, 0x01, 0x11, 0x01, 0x00,
    0x00, 0x80, 0xc0, 0xf0, 0xfc, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xcc, 0x80, 0x00, 0x00, 0x00, 0x07, 0x3f, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0x02, 0x10, 0x00,
    0xf8, 0xff, 0xff, 0xff, 0x1f, 0x1f, 0x00, 0x00, 0x00, 0xf8, 0xf0, 0xf0, 0xe0, 0xe0, 0x90,
    0x01, 0x03, 0x1f, 0x0f, 0x0f, 0x1f, 0x3f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0x0f, 0x03, 0x00, 0x80, 0xc0, 0xe0, 0xc0, 0xc0, 0x80, 0x00, 0x01,
    0x1f, 0x3f, 0x7f, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0x7f, 0x3f, 0x0f, 0x03, 0x01, 0x00, 0x80, 0xe0, 0xf8,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
    0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff, 0xf0, 0xe0, 0xc0, 0xc0, 0x80, 0x03,
    0x0f, 0x0f, 0x0f, 0x1f, 0x1f, 0x1f, 0x1f, 0x1f, 0x0f, 0x07, 0x83, 0xc0, 0xe0, 0xf8, 0xfc,
    0xff, 0xff, 0xff, 0xff, 0xfc, 0x00, 0x80, 0x80, 0xe8, 0xff, 0xff, 0xff, 0xff, 0xff, 0xff,
```

Ln 22, Col 101 Tab Size: 4 UTF-8 LF {} C
TestScreen | Energia 1.6.10E18
Dosya Düzenle Taslak Araçlar Yardım
TestScreen

```
#include "Iseplogo128.h"
void setup() {
  // put your setup code here, to run once:

InitI2C();
InitScreen();

Display(motif);                          // affichage de l'image décrite dans le tabelau de donnée motif.h
//DisplayString(0,5,"BONJOUR");            // Ecriture sur la 5ème ligne
//DisplayString(0,7,"ISEP");  // Ecriture sur la dernière ligne.



// void test_fft(); // La fonction test_fft disponible dans le fichier fft.ino montre comment utiliser la fft

}

void loop() {

  // put your main code here, to run repeatedly:

}
```

```
Yükleme tamamlandı.
        Setting PC to entry point.: 70%
Running...
```

2nd

Replace the code into the sample file" iseplogo128.h", and then run in PUTTY.

Team

NAME

Title

NAME

Title

LIU YANG

Aeronautics

NAME

Title