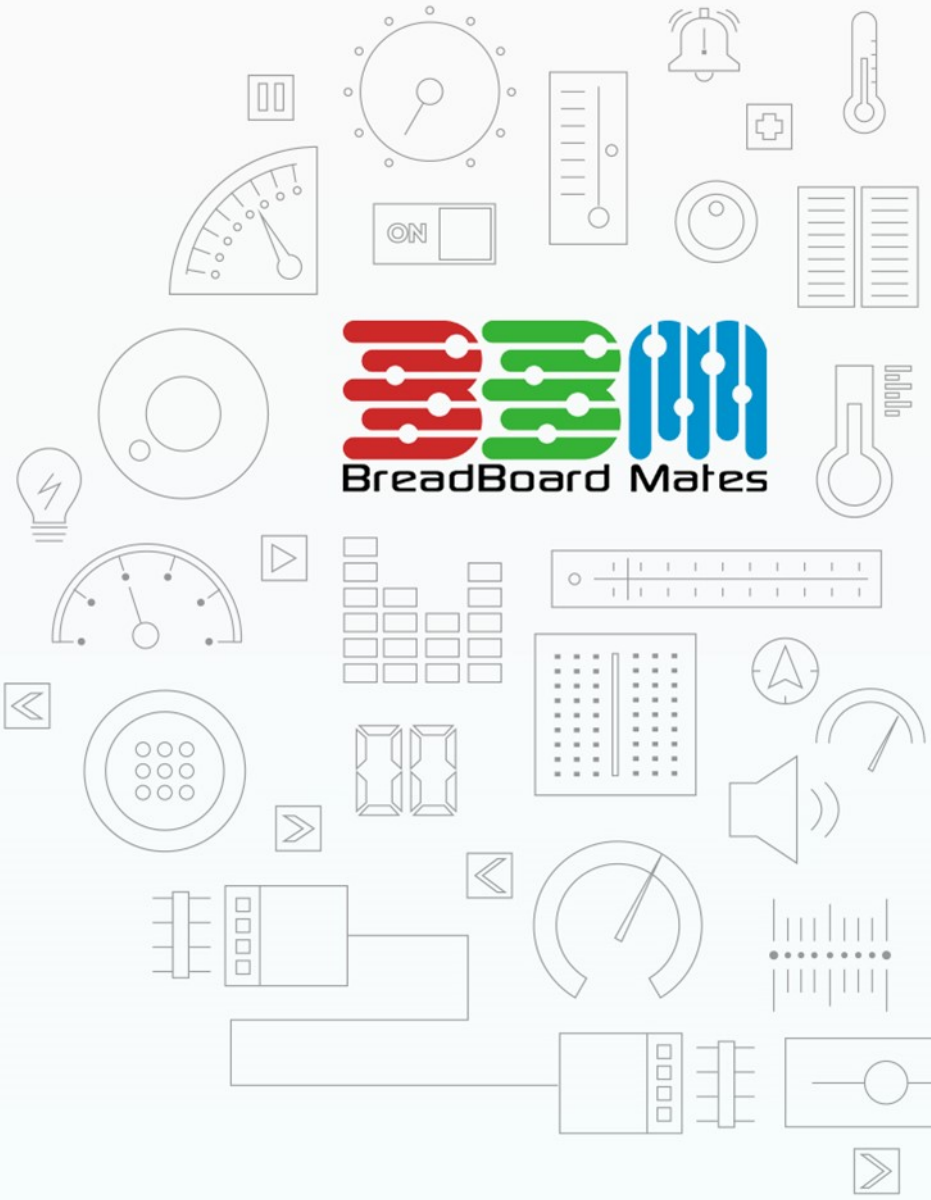


PROJECT

Simulated Animated Text from Arduino

Content may change at any time. Please refer to the resource centre for latest documentation.

Contents

Introduction

Arduino's are very common microcontroller boards used to study and design programmable electronics. It is often used with multiple peripherals such as buttons, sliders, sensors and motors.

Together with a TIMI acting as a small fancy display, Arduino boards become a lot more powerful and interesting to use in prototyping.

This project showcases a TIMI-96 module controlled by an Arduino Uno to display strings in a PrintArea. This also shows how to clear the PrintArea.

Requirements

To proceed with the project, the following are required.

Hardware

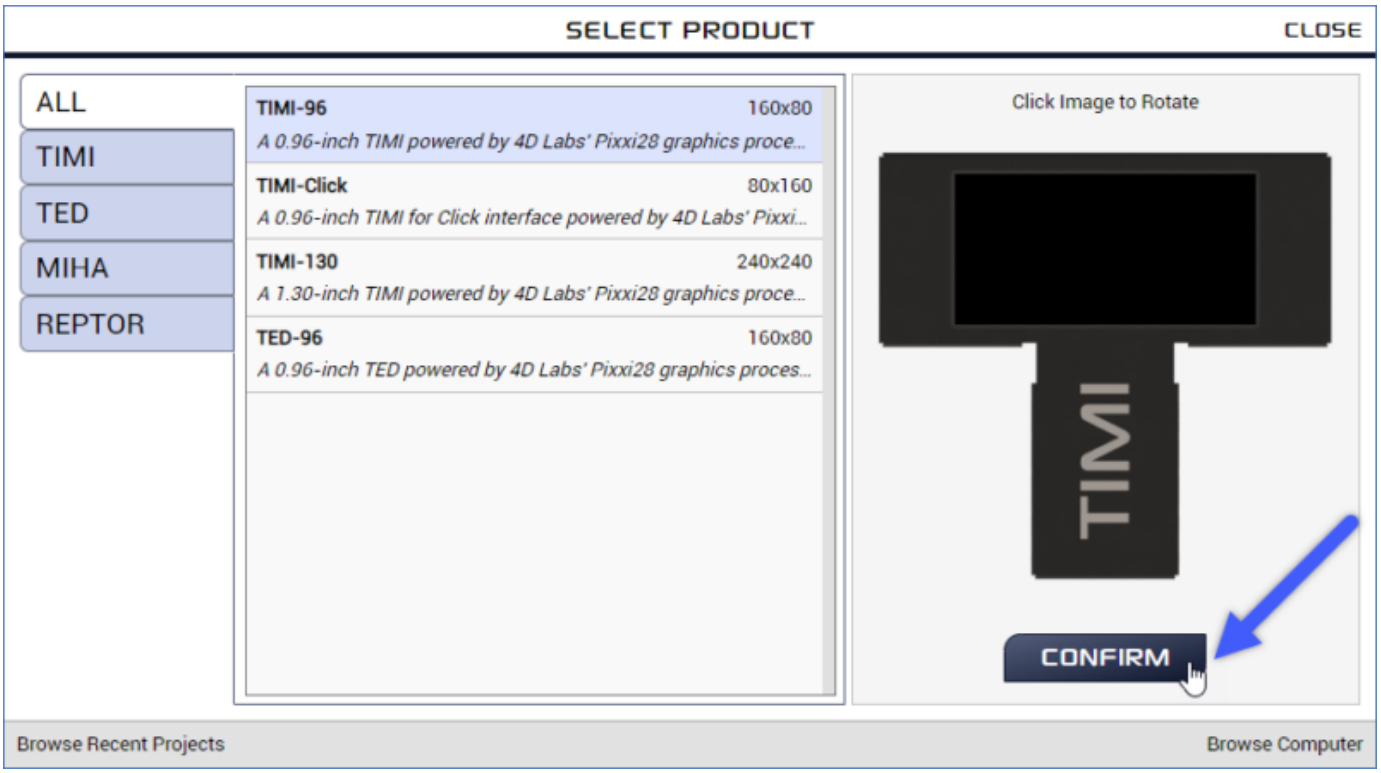
- TIMI-96
- Mates Programmer
- USB Type A to microUSB cable (for the Mates Programmer)
- USB Type A to Type B cable (for the Arduino, replace as necessary)
- Connecting Wires
- Arduino Uno
- Breadboard

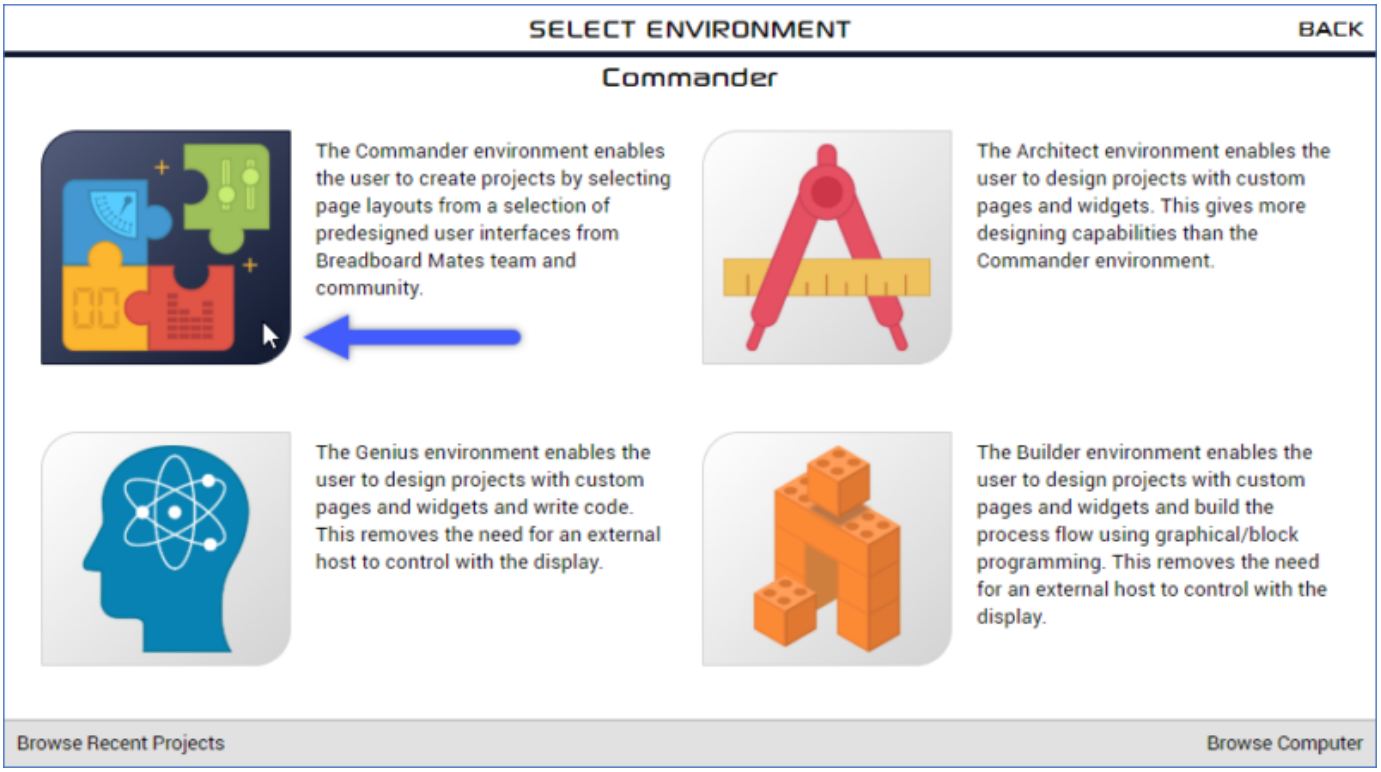
Software

- Mates Studio
- Arduino IDE

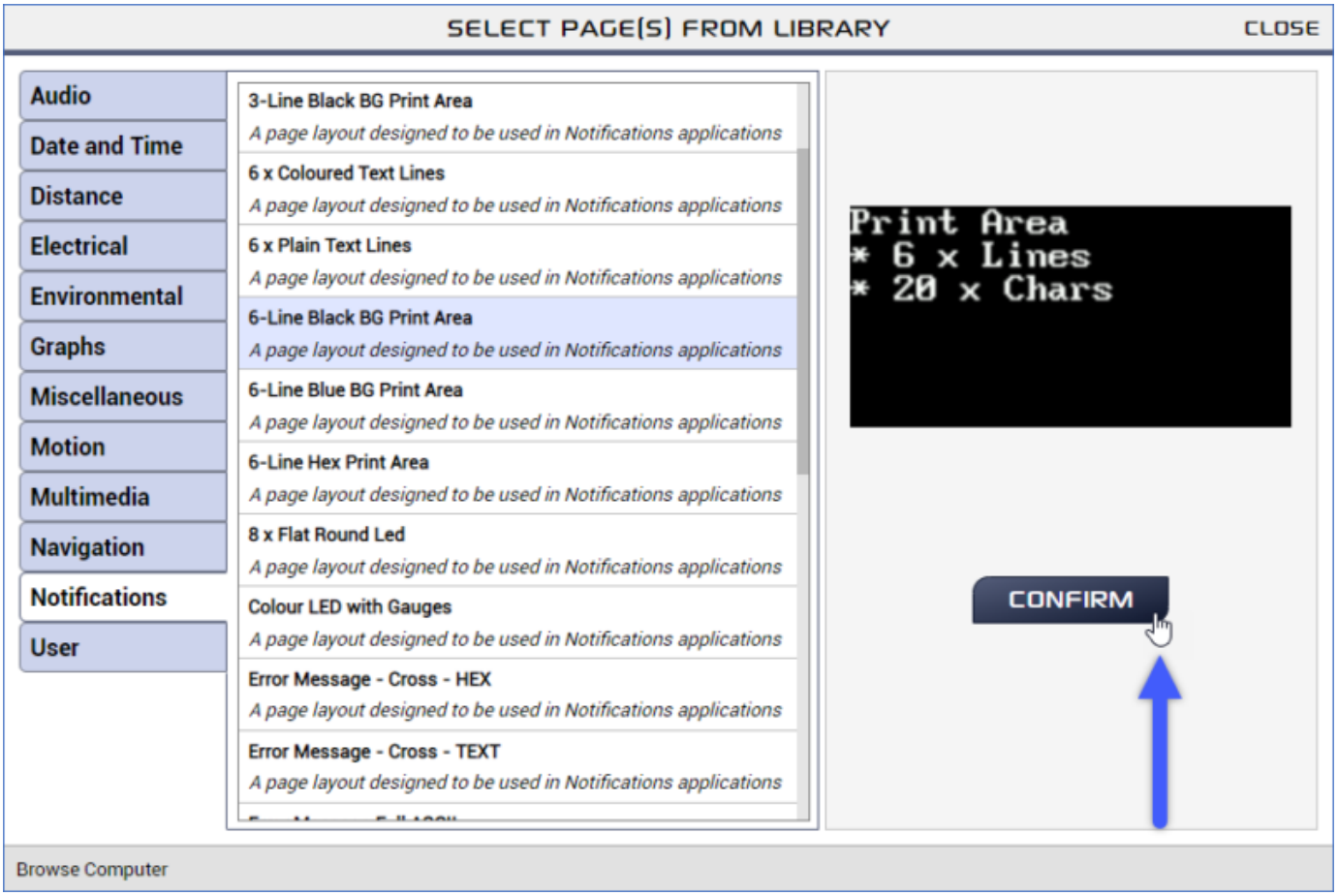
Graphics Design

Step 1: Open *Mates Studio* and create a *Commander* project for *TIMI-96* with *Reversed Landscape* orientation

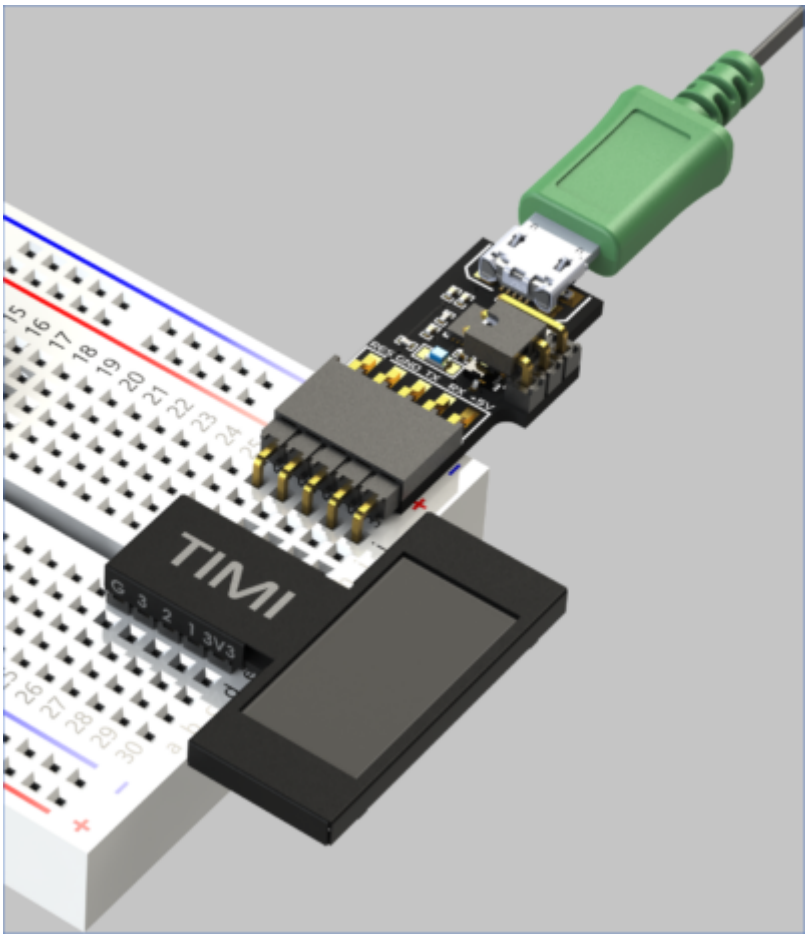




Step 2: Browse the library for appropriate page designs. For this project, *6-Line Black BG Print Area* page under *Notifications* category was used.

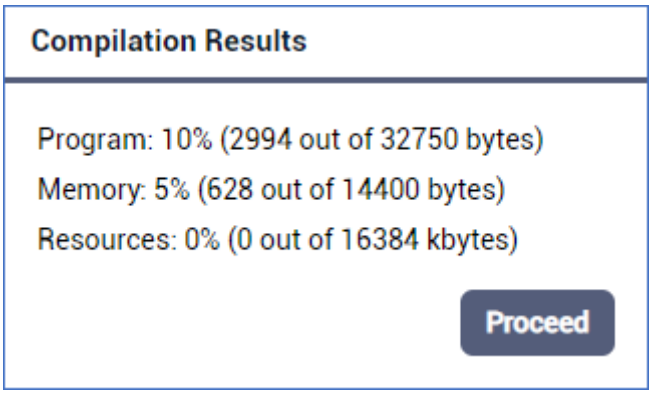


Step 3: After finalizing the design, connect TIMI-96 to your computer



Step 4: Upload the project to the appropriate COM port

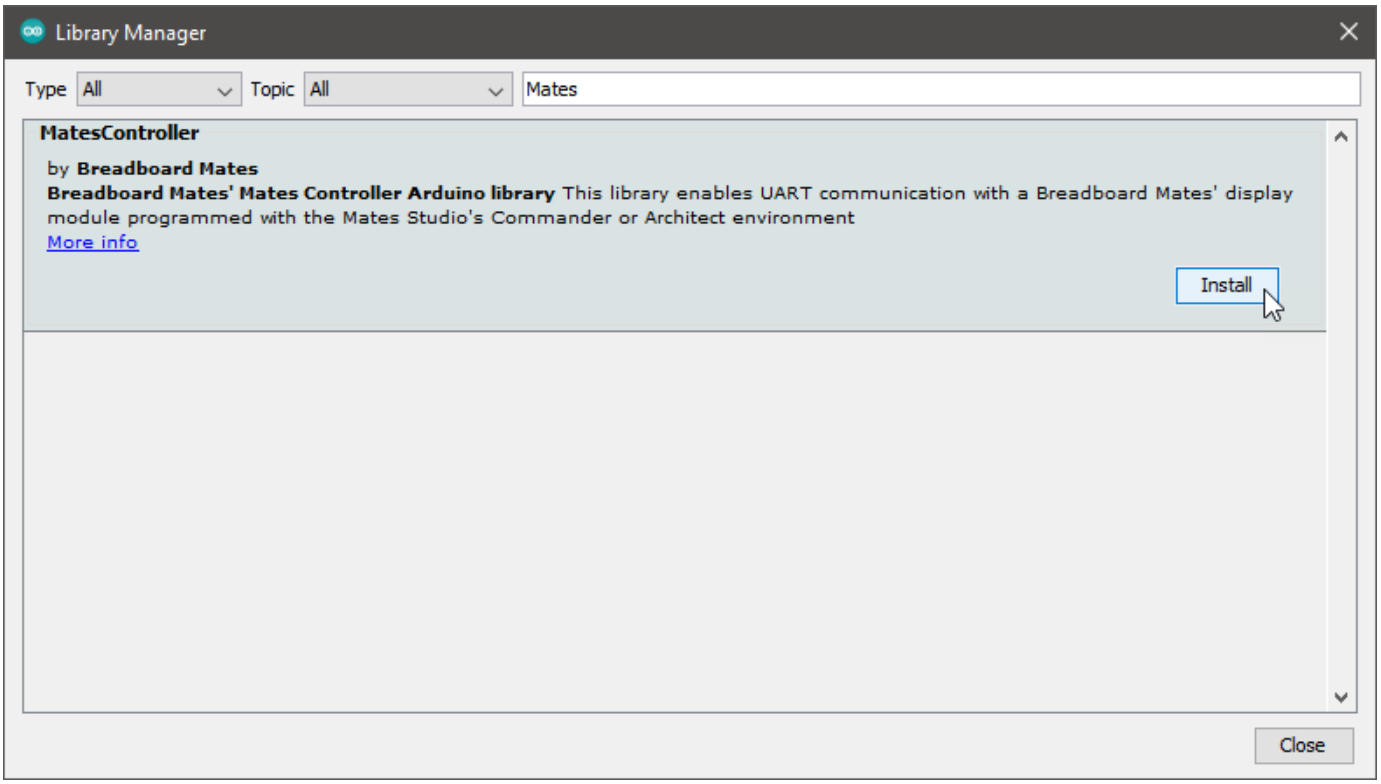
Step 5: When prompted, click *Proceed* to continue with upload.



> **Note**
>
> It is recommended that the graphics design is finalized before moving to the next steps when working on a project*

Programming the Arduino

Step 1: Install the *MatesController* library using Arduino's *Library Manager*.



Step 2: Include *MatesController.h* to your project.

```
#include "MatesController.h"
```

Step 3: Create a *MatesController* instance named *mates*.

```
MatesController mates = MatesController(Serial);
```

This will initialize the *MatesController* instance to the default reset pin 4 using a LOW pulse.

Step 4: (Optional) Create a function for toggling the built-in LED of the Arduino board. This can be used for debugging or showing errors if the Serial monitor can't be used.

```
int errLedStatus = LOW;
void ErrorLed_Toggle() {
  errLedStatus = ~errLedStatus;
  digitalWrite(LED_BUILTIN, errLedStatus);
}
```

Step 5: (Optional) At the beginning of the setup function, set the built-in LED pin to OUTPUT and set it to LOW.

```
pinMode(LED_BUILTIN, OUTPUT);
digitalWrite(LED_BUILTIN, errLedStatus);
```

Step 6: To start using the MatesController instance, use the `begin` function

```
mates.begin();
```

This will initialize the Serial UART at the default baudrate of 9600

Step 7: (Optional) The `begin` function can be enclosed in an if condition to handle initialization errors.

```
if (!mates.begin()) {
  // Display didn't send ready signal in time
  while (1) {
    ErrorLed_Toggle();
    delay(100);
  }
}
```

Step 8: Prepare a constant string to send to the PrintArea.

```
char str[2];
unsigned long lastUpdate;
```

Step 9: Some additional variables are required for this project. Prepare a two character array to store a single character string and an unsigned long variable to store the system time of the last update.

```
char str[2];
unsigned long lastUpdate;
```

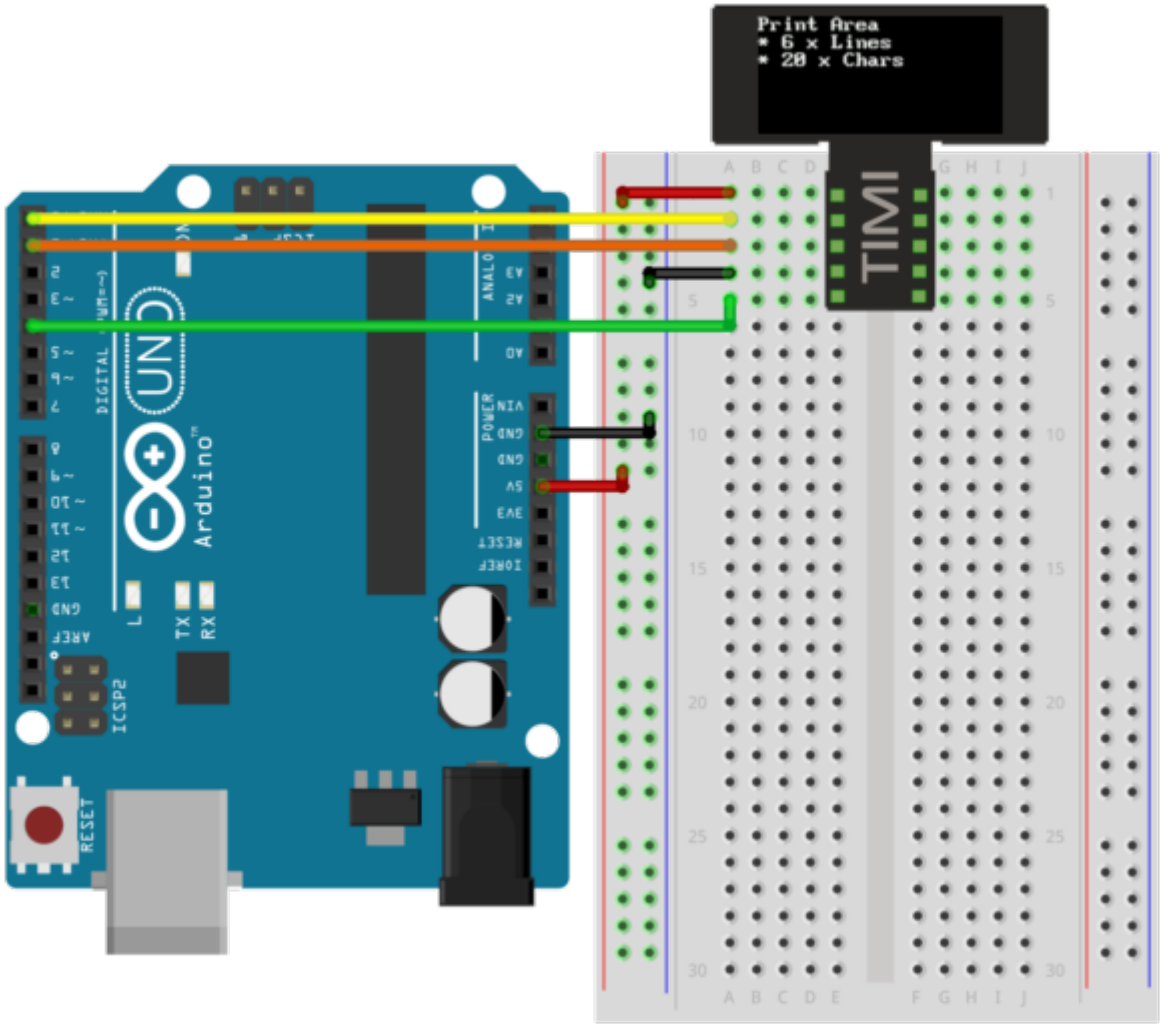
Step 10: In the loop function, the values are simulated and sent to TIMI as necessary.

```
void loop() {
  if (millis() - lastUpdate >= 3000) {

    mates.clearPrintArea(0);

    for (uint8_t i = 0; i < strlen(msg); i++) {
      str[0] = msg[i];
      str[1] = 0;
      mates.appendToPrintArea(0, str);
      delay(50);
    }

    lastUpdate = millis();
  }
}
```

As shown, the code utilizes a non-blocking delay to execute the update every 3 seconds. Each update starts by clearing the PrintArea followed by printing the string one character at a time.

Running the Project

After designing the user interface for TIMI and writing code for the Arduino and programming them, it is time to connect the devices together. Follow the diagram below for the connection between TIMI and Arduino.



Finally, supply power to the Arduino and observe the behavior of the project.

Downloadable Resources

The Mates Studio – Commander project is included in the MatesController library. It is available under the extras folder of the library. You can find it in (if the library was installed using Arduino Library Manager):

```
C:\Users\%USERNAME%\Documents\Arduino\libraries\MatesController\extras\Print Strings.mates
```

Here are the links to the software applications, libraries and completed project files.

- Mates Studio
- Arduino IDE
- Arduino Mates Controller Library
- Project Files