APPLICATION NOTES

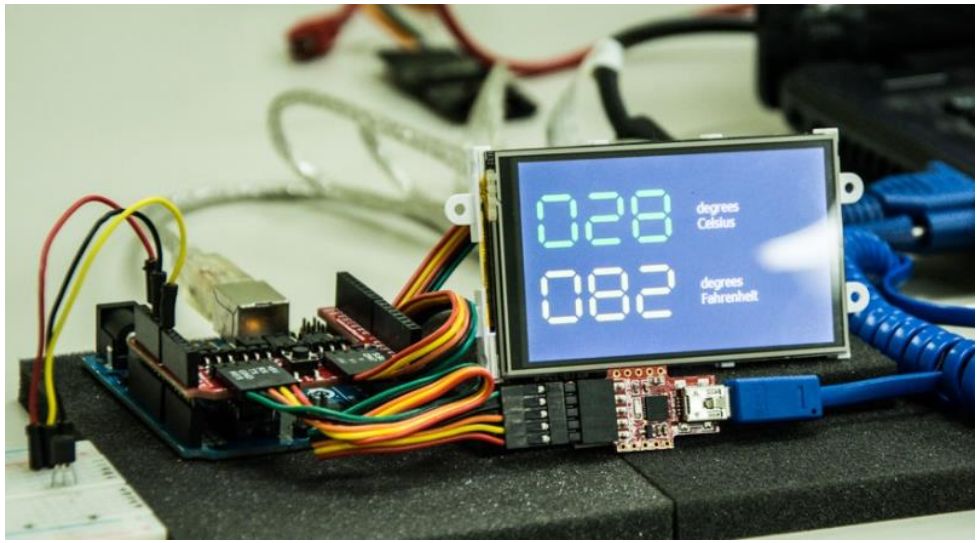
4D SYSTEMS
TURNING TECHNOLOGY INTO ART

ViSi-Genie Displaying Temperature Values from an Arduino Host

DOCUMENT DATE: **4th April 2019**
DOCUMENT REVISION: **1.1**

Description

This application note explains how to use a 4D display module in displaying temperature values received from a host controller. The host is an AVR-ATmega328-microcontroller-based Arduino Uno board, to which an LM35 temperature sensor is connected. The host can also be an Arduino Mega 2560 or Due. Ideally, the applications described in this document should work with any Arduino board with at least one UART serial port. See specifications of Aduino boards here.



This application note requires:

- Any of the following 4D Picaso and gen4 Picaso display modules:

gen4-uLCD-24PT	gen4-uLCD-28PT	gen4-uLCD-32PT
uLCD-24PTU	uLCD-32PTU	uVGA-III

  and other superseded modules which support the ViSi Genie environment

- The target module can also be a Diablo16 display

gen4-uLCD-24D series	gen4-uLCD-28D series	gen4-uLCD-32D series
gen4-uLCD-35D series	gen4-uLCD-43D series	gen4-uLCD-50D series
gen4-uLCD-70D series		
uLCD-35DT	uLCD-43D series	uLCD-70DT

  Visit www.4dsystems.com.au/products to see the latest display module products that use the Diablo16 processor. The display module used in this application note is the uLCD-32PTU, which is a Picaso display. This application note is applicable to Diablo16 display modules as well.
- 4D Programming Cable / uUSB-PA5/uUSB-PA5-II
  for non-gen4 displays(uLCD-xxx)
- 4D Programming Cable & gen4-PA, / gen4-IB / 4D-UPA
  for gen4 displays (gen4-uLCD-xxx)
- micro-SD (µSD) memory card
- Workshop 4 IDE (installed according to the installation document)
- Any Arduino board with a UART serial port

- 4D Arduino Adaptor Shield (optional) or connecting wires
- Arduino IDE
- When downloading an application note, a list of recommended application notes is shown. It is assumed that the user has read or has a working knowledge of the topics presented in these recommended application notes.

Content

Application Overview

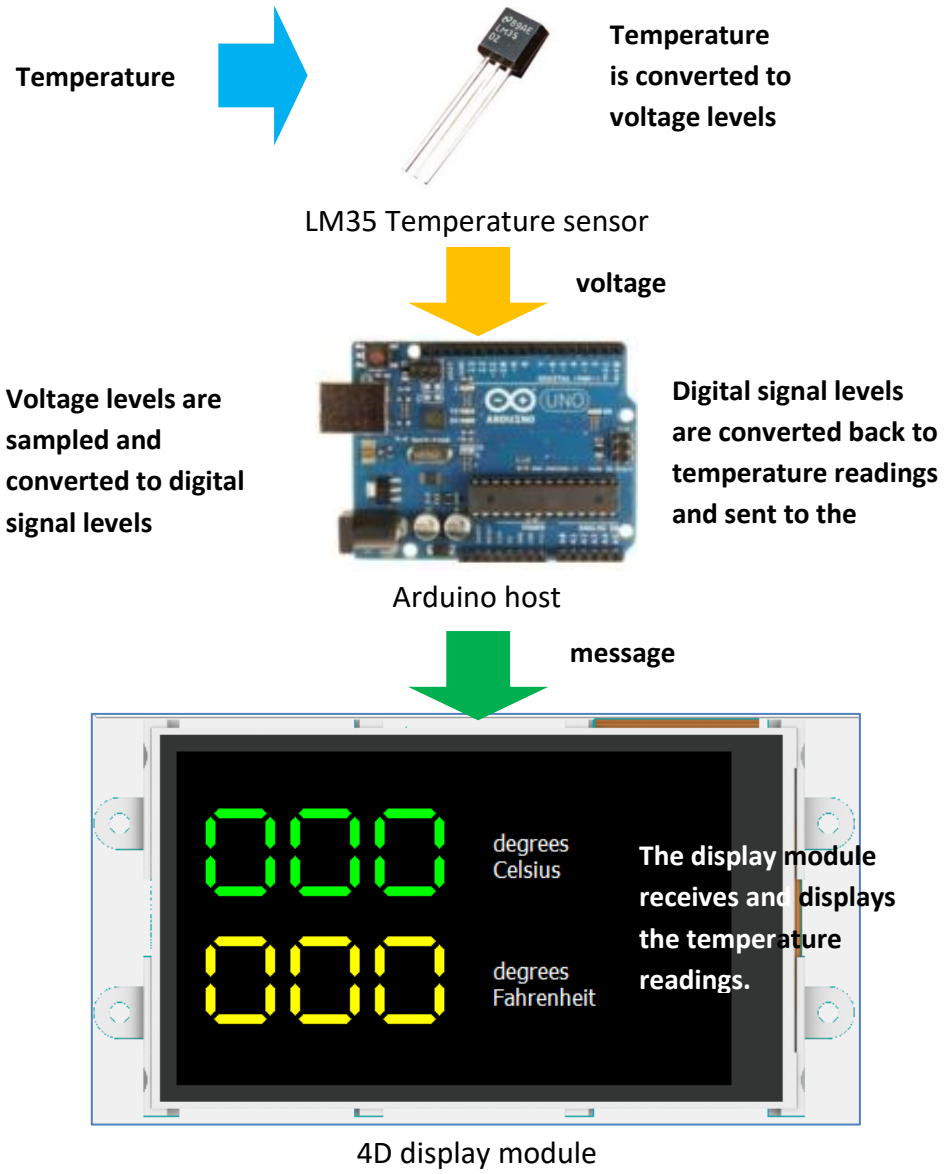
The application developed in this document works in a manner illustrated in the diagram shown on the next page.

First, temperature is converted to an electrical signal (voltage to be exact) by an LM35 temperature sensor. The LM35 temperature sensor used in this application has a scale factor of 10mV per degree Celsius. It is configured to sense temperature values from 2 to 150 degrees Celsius. Users who intend to read temperature values outside this range may need to consult the datasheet for more information.

Next, the voltage output of the LM35 temperature sensor is sampled by pin A0 of the Arduino Uno. A0 is one of the six pins of the Arduino Uno capable of performing analogue to digital signal level conversion – or ADC. Given the scale factor, the voltage level output of the LM35 temperature sensor can now be converted back to a temperature reading – both in degrees Celsius and degrees Fahrenheit. These temperature values are now sent to the display module.

The Arduino host is programmed in the Arduino IDE to perform ADC, convert digital signal levels to temperature readings, and send the temperature readings to the display module. The display module, on the other hand, is programmed in Workshop ViSi Genie environment to display the temperature readings.

This application note comes with a ViSi Genie program and an Arduino sketch. The process of creating the ViSi Genie program is first shown. Then the flow of the Arduino sketch is discussed. The sketch can be used to develop more complex applications. The last section shows how the display, Arduino host, and LM35 temperature sensor are connected.



Setup Procedure

For instructions on how to launch Workshop 4, how to open a ViSi-Genie project, and how to change the target display, kindly refer to the section "**Setup Procedure**" of the application note:

ViSi Genie Getting Started – First Project for Picaso Displays (for Picaso)
or
ViSi Genie Getting Started – First Project for Diablo16 Displays (for Diablo16).

Create a New Project

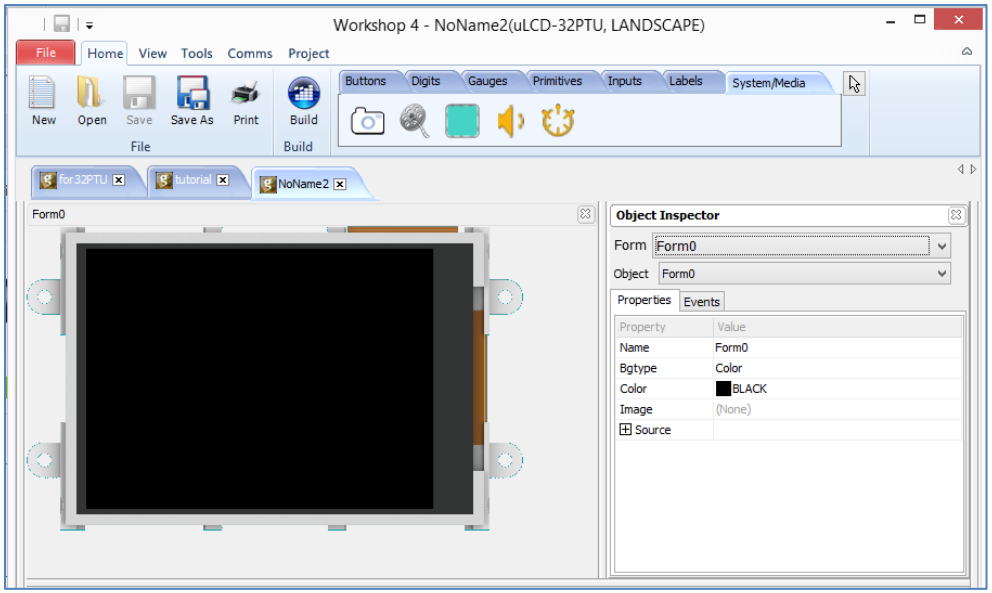
Create a New Project

For instructions on how to create a new ViSi-Genie project, please refer to the section "**Create a New Project**" of the application note

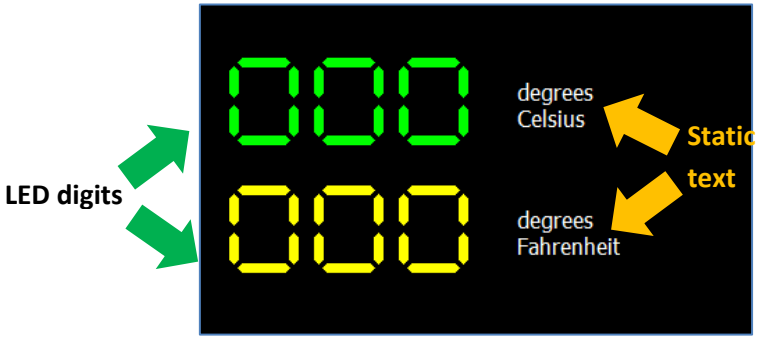
ViSi Genie Getting Started – First Project for Picaso Displays (for Picaso)
or
ViSi Genie Getting Started – First Project for Diablo16 Displays (for Diablo16).

Design the Application

Everything is now ready to start designing the project. **Workshop 4** displays an empty screen, called **Form0**. A **form** is like a page on the screen. The form can contain **widgets** or **objects**, like sliders, displays or keyboards. Below is an empty form.

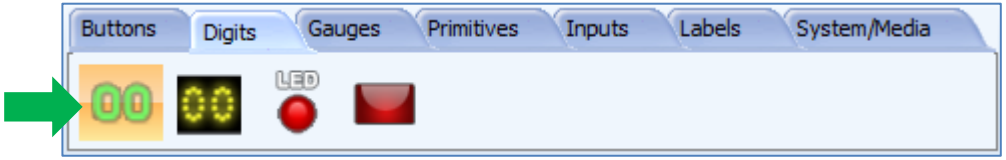


At the end of this section, the user will able to create a form with four objects. The final form will look like as shown below, with the labels excluded.

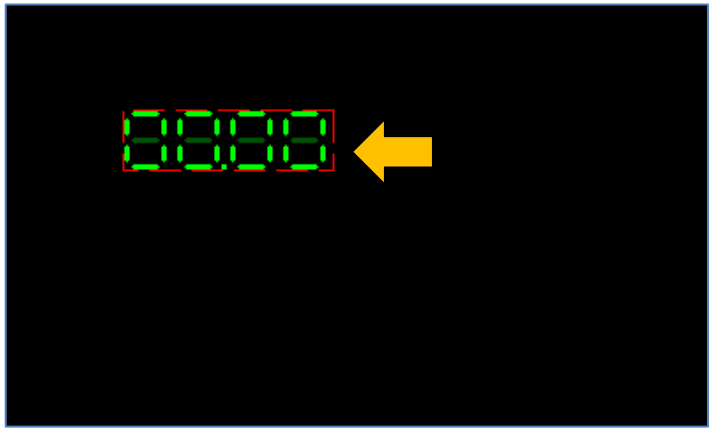


Create a LED Digits Object

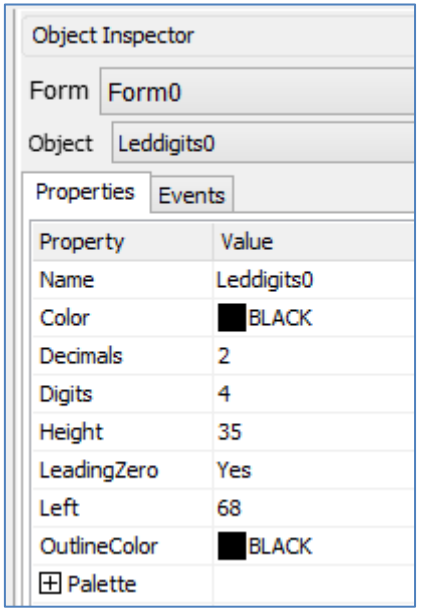
The **LED digits** object will display values received from the Arduino host. To add a LED digits object, go to the **Digits** pane and select the first icon.



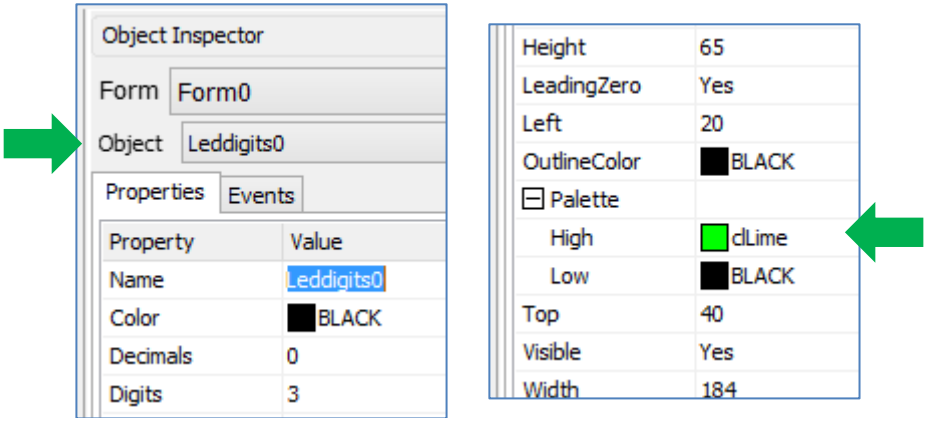
Click on the **WYSIWYG** (What-You-See-Is-What-You-Get) screen to place a LED digits object. The WYSIWYG screen simulates the actual appearance of the display module screen.



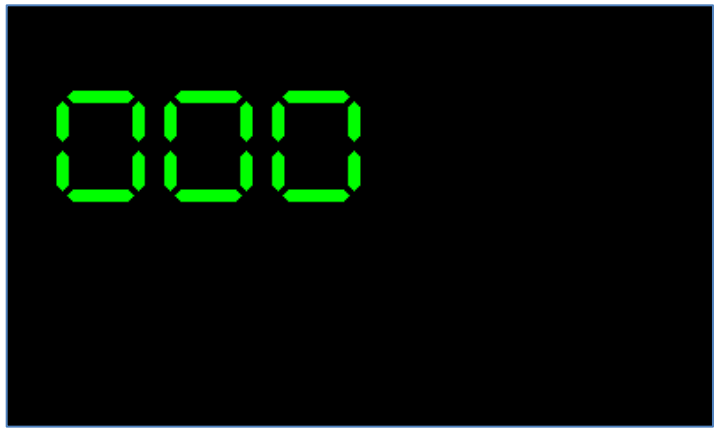
The object can be dragged to any desired location and resized to the desired dimensions. The **Object Inspector** on the right part of the screen displays all the properties of the newly created LED digits object named **Leddigits0**.



Feel free to experiment with the different properties. To know more about digital display objects, refer to [ViSi-Genie Digital Displays](). Now set the following properties as indicated below.

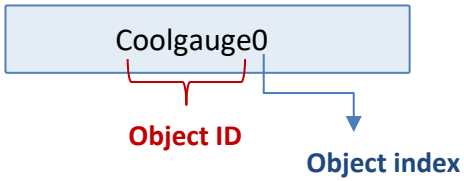


The WYSIWYG screen is updated. The user can also set the preferred height and width of the object.



Naming of Objects

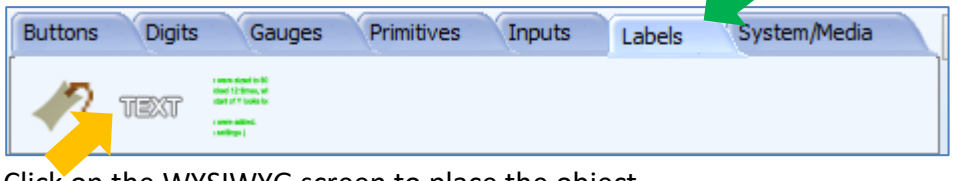
Naming is important to differentiate between objects of the same kind. For instance, suppose the user adds another LED digits object to the WYSIWYG screen. This object will be given the name Leddigits1 – it being the second LED digits in the program. The third LED digits object will be given the name Leddigits2, and so on. An object's name therefore identifies the object's kind and unique index number. It has an ID (or type) and an index.



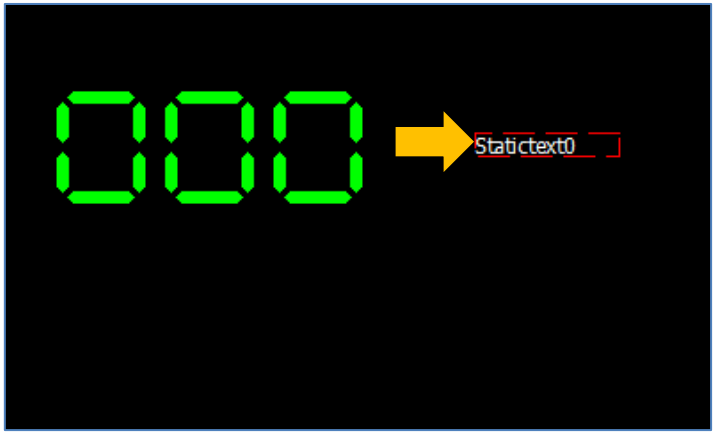
It is important to take note of an object's ID and index. When programming in the Arduino IDE, an object's status can be polled or changed if its ID and index are known. The process of doing this will be shown later.

Add a Static Text

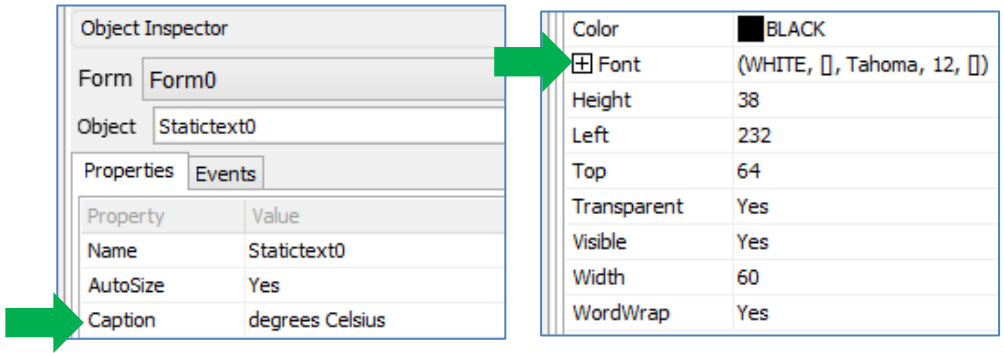
Static text objects are useful for labelling purposes. As the name implies, the status of these objects cannot be changed when the program runs. To add a static text, go to the Labels pane and click on the static text icon.



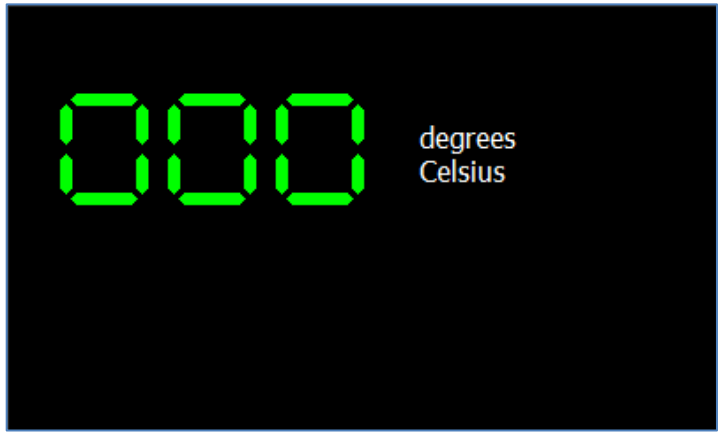
Click on the WYSIWYG screen to place the object.



In the Object Inspector, change the caption to degrees Celsius, and increase the font size.



When done, the WYSIWYG screen should look similar to the one shown below.



Repeat the process described above to create the LED digits and static text objects for the temperature readings in degrees Fahrenheit.

Build and Upload the Project

For instructions on how to build and upload a ViSi-Genie project to the target display, please refer to the section "**Build and Upload the Project**" of the application note

ViSi Genie Getting Started – First Project for Picaso Displays (for Picaso)
or
ViSi Genie Getting Started – First Project for Diablo16 Displays (for Diablo16).

The uLCD-32PTU and/or the uLCD-35DT display modules are commonly used as examples, but the procedure is the same for other displays.

Writing the Host Code

A thorough understanding of the application note **ViSi-Genie Connecting a 4D Display to an Arduino Host** is required before attempting to proceed further beyond this point. **ViSi-Genie Connecting a 4D Display to an Arduino Host** provides all the basic information that a user needs to be able to get started with ViSi-Genie and Arduino. The following is a list of the topics discussed in **ViSi-Genie Connecting a 4D Display to an Arduino Host**.

- How to download and install the ViSi-Genie-Arduino library
- How to open a serial port for communicating with the display and how to set the baud rate
- The genieAttachEventHandler() function
- How to reset the host and the display
- How to set the screen contrast
- How to send a text string
- The main loop
- Receiving data from the display
- The use of a non-blocking delay in the main loop
- How to change the status of an object
- How to know the status of an object
- The user's event handler

Discussion of any of these topics is avoided in other ViSi-Genie-Arduino application notes unless necessary. Users are encouraged to read **ViSi-Genie Connecting a 4D Display to an Arduino Host** first.

The Main Loop – Writing Data to the Display

The line

```
tempC = ((5.0 * analogRead(sensorPin) * 100.0) / 1024) + 2;
```

takes a digital signal level sample from sensorPin (A0) and converts it to a temperature reading. The line performs the conversion shown below.

$$tempC = sensorPin\ levels\ x\ \frac{5\ volts}{1024\ levels}\ x\ \frac{1\ degree\ Celsius}{10\ mVolts}$$

ADC resolution of the Arduino Uno

Scale factor of LM35

An additional 2 degrees Celsius is added

```
tempC = ((5.0 * analogRead(sensorPin) * 100.0) / 1024) + 2;
```

to account for the fact that the LM35 temperature sensor used in this example has a voltage output starting at 0 mV and increasing at 10 mV per degree from 2 to 150 degrees Celsius. Consult the datasheet for your LM35 temperature sensor for more information.

The lines

```
genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x00, tempC);
```

and

```
genie.WriteObject(GENIE_OBJ_LED_DIGITS, 0x01, tempF);
```

send the temperature values to the display module.

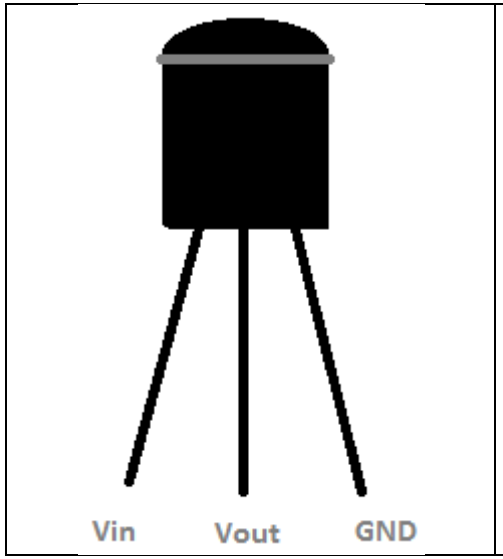
Set Up the Project

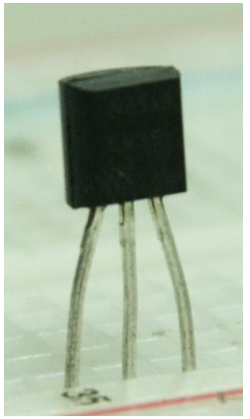
Refer to the section "**Connect the Display Module to the Arduino Host"** of the application note "**ViSi-Genie Connecting a 4D Display to an Arduino Host**" for the following topics:

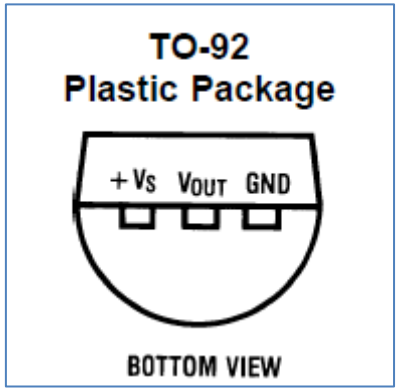
- Using the New 4D Arduino Adaptor Shield (Rev 2.00)
  - Definition of Jumpers and Headers
  - Default Jumper Settings
  - Change the Arduino Host Serial Port
  - Power the Arduino Host and the Display Separately
- Using the Old 4D Arduino Adaptor Shield (Rev 1)
- Connection Using Jumper Wires
- Changing the Serial port of the Genie Program
- Changing the Maximum String Length

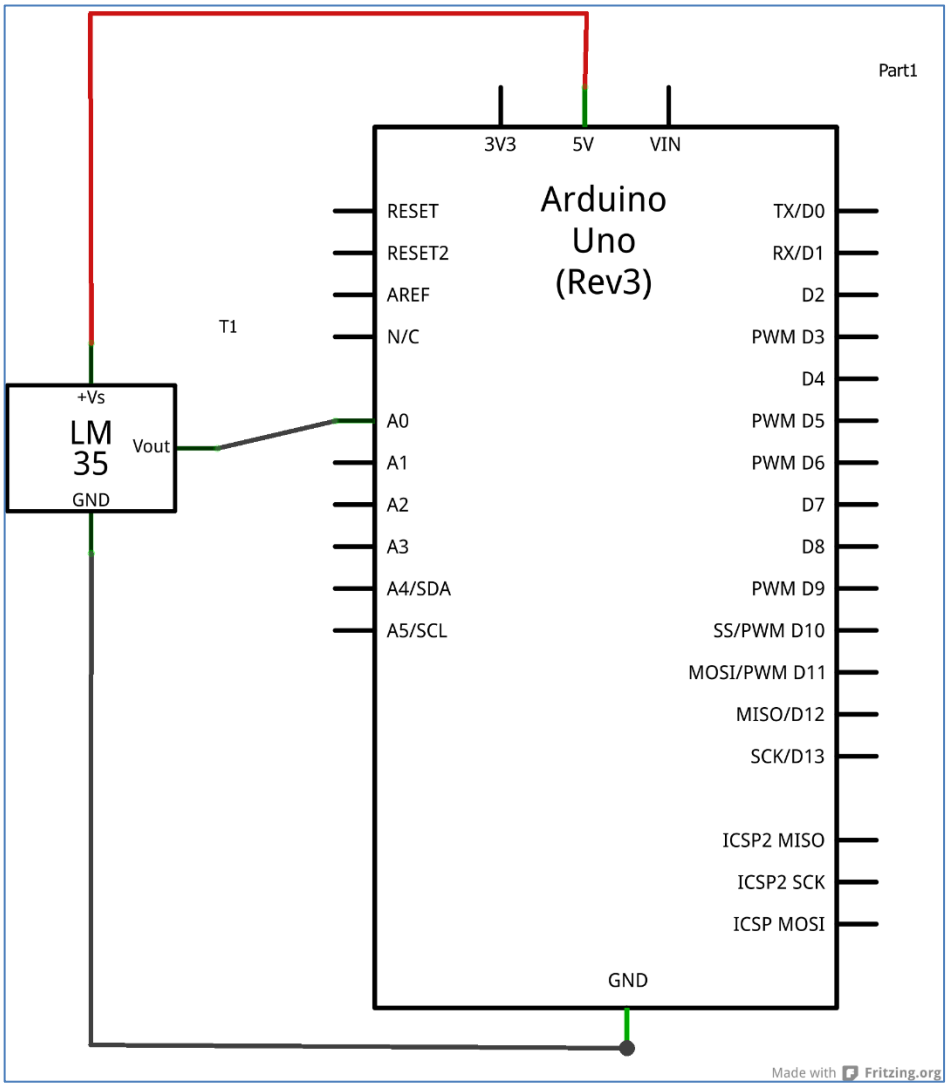
Connect the LM35 Temperature Sensor to the Arduino

Refer to the datasheet for detailed information.

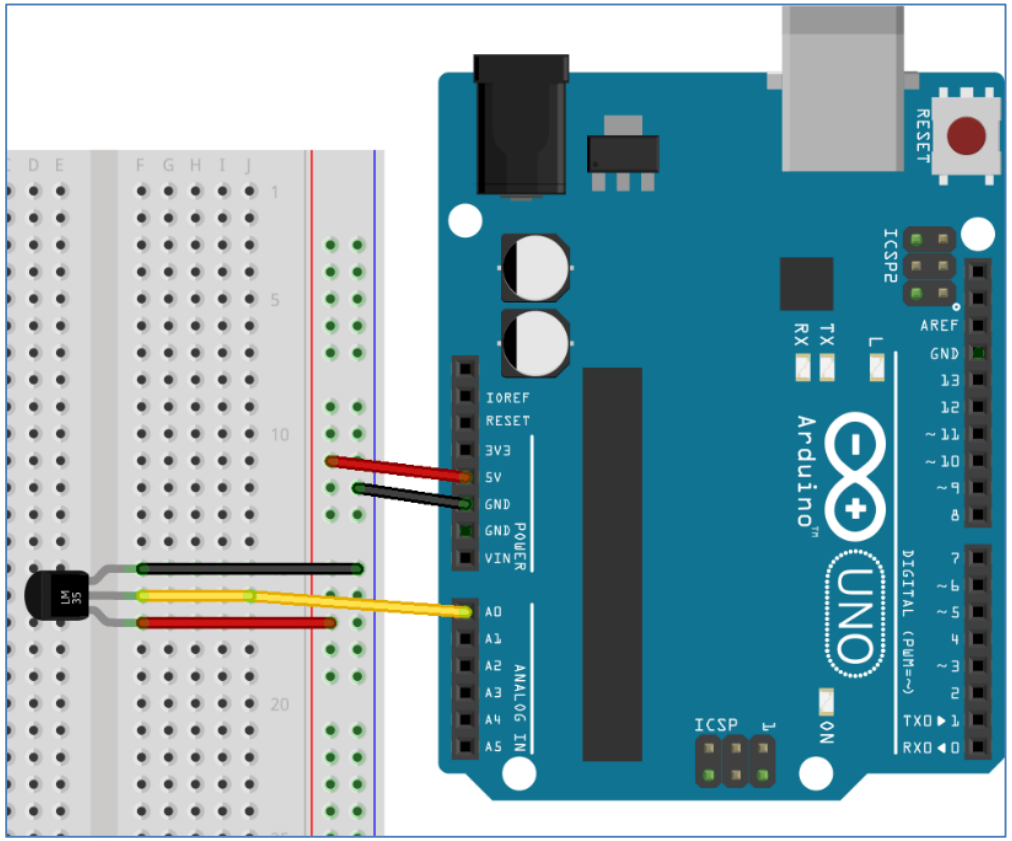








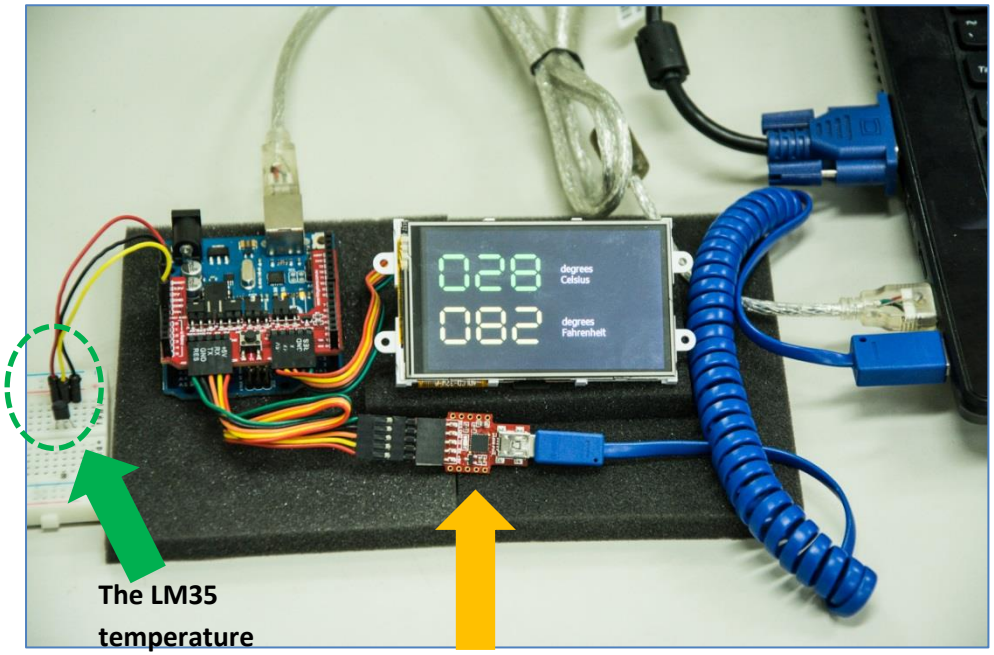
Schematic for LM35-to-Arduino-Uno connection (made with Fritzing)



Breadboard layout for LM35- to-Arduino-Uno connection (made with Fritzing)

The Complete Project

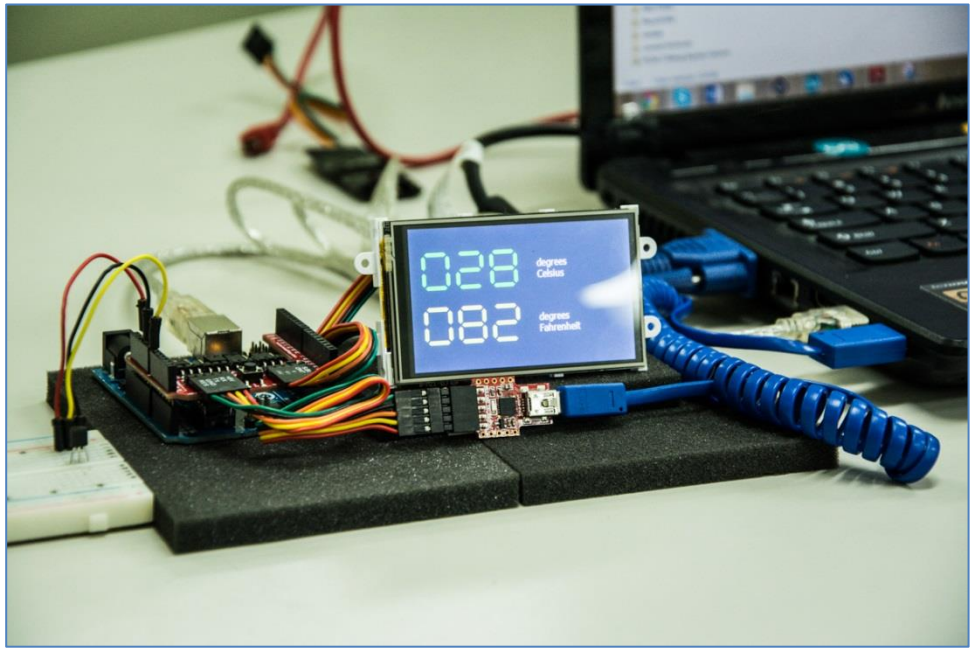
At room temperature:

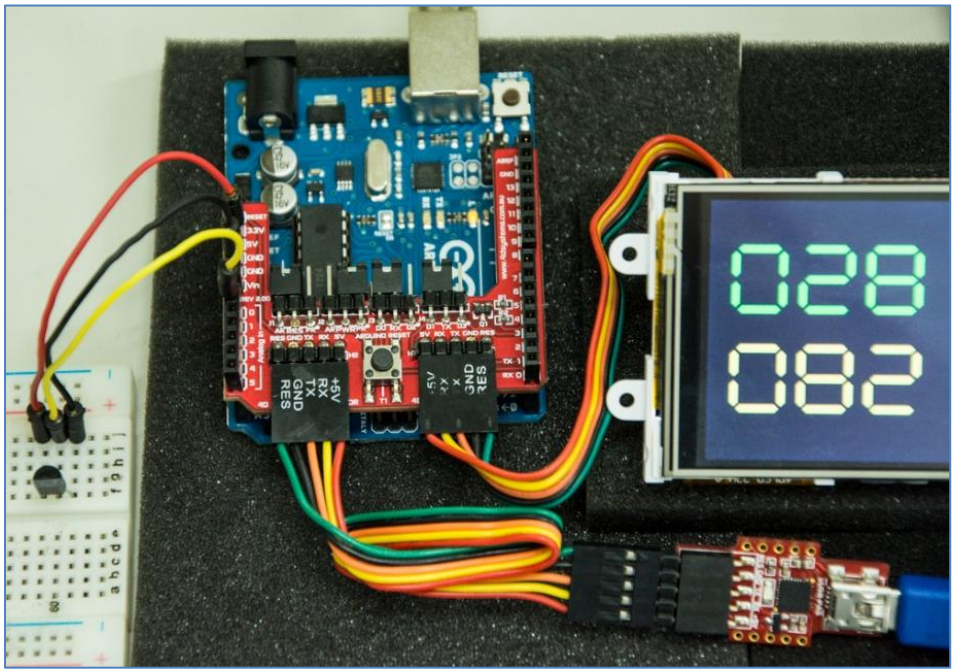


The LM35 temperature sensor

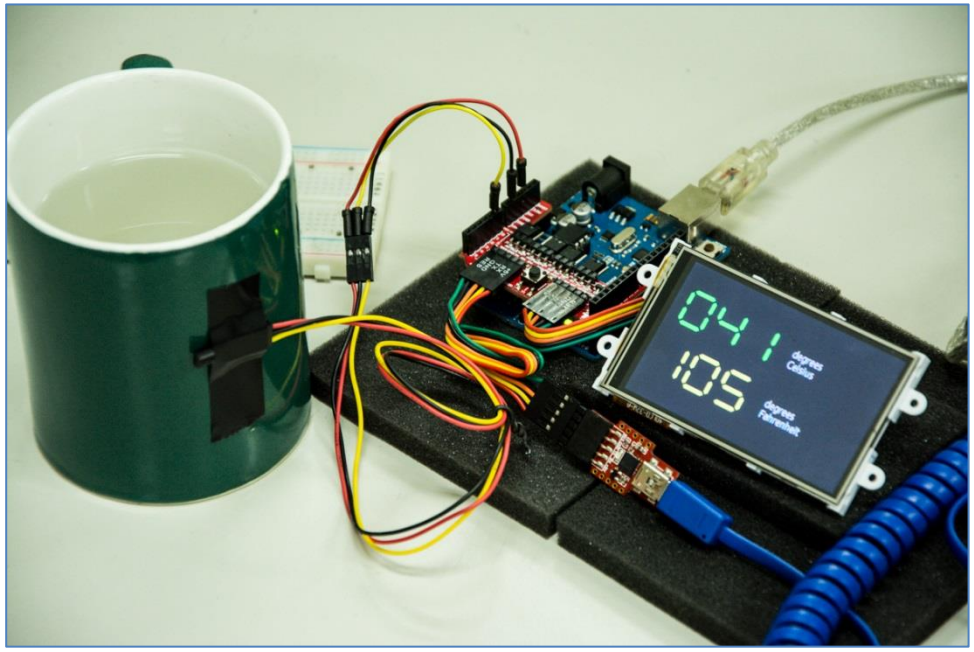
The uUSB - PA5 programming adapter

The uUSB-PA5 is used only to power the display.



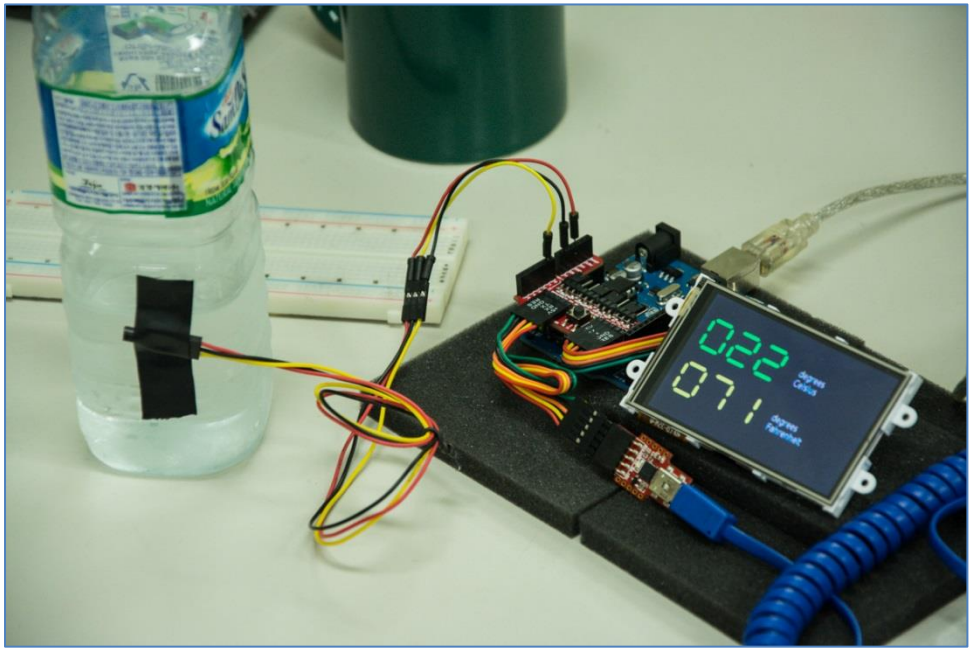


Temperature of a warm cup of water:



The LM35 temperature sensor is removed from the breadboard and connected to the Arduino Uno using a three-way cable.

Temperature of a cold bottle of water:



Proprietary Information

Disclaimer of Warranties & Limitation of Liability