Floating Point (FP) Display for the UNO V2

Introduction

The UNO floating point (FP) display pcb plugs into the UNO. It accepts numerical strings and binary numbers from the UNO and displays them on an 8 digit display.

Two projects are posted:

FP_display_V2, developed using Studio 7. This runs on an ATtiny 1606 on the display pcb.

UNO_template_V2. This contains subroutines required by a UNO application to communicate with the display pcb.

For basic background information the reader is referred to version 1 of this project.

The purpose of V2

This project has been developed to serve as an introduction to the ATtiny 1606 family of devices. The ATtiny 1606 has been chosen because it is a 20 pin device ideally suited to driving an 8 digit display. In addition the 1606 family:

Are relatively new devices and benefit from recent developments affecting the way in which they are programmed and code is written.

Come in a SOIC package which is amenable to basic hand soldering and is an excellent compromise between DIP and smaller surface mount packages.

Programming the 1606

The 1606 has provided a means of testing my UPDI programmer (published on 27 March 2021). Programming the display pcb device is much simpler that it was for V1 where two devices were required. Programmer code and template have been merged into a single sketch. They can be separated out when development is complete.

Missing features

This project does not include all the features provided by V1. Data input using the IO, and overflow/underflow are not dealt with and only CA displays are supported. The aim here is to set up a basic one wire link between the UNO and 1606 over which numbers can be transferred in binary and string form. The 1606 does offer a 1 wire UART however a firm ware implementation has been adopted. For a basic application such as this it probably offers more flexibility.

Clock accuracy

The ATtiny1606 is clocked from an internal oscillator. A calibration register is used to fine tune this oscillator. In addition configuration fuses are provided giving the precise oscillator frequency. Tests have revealed that communication between the UNO and 1606 is fine for four consecutive values of the fine tuning register. It is therefore hoped that everything will be fine with the oscillator in its default configuration.
Code is provided to adjust the fine tuning register if necessary (possibly due to oscillator accuracy or SW latency). However if this is used Studio 7 will be required to recompile FP_display_V2.

Driving the combined programmer/template application

A typical Bray++ terminal session is shown below. In general it is not necessary to verify programming. (i.e. the programmer either works all the time or none of it.)

Extricating the UNO template

Remove the forward slash from the end of line:

/******….…....Programmer code starts here and can be removed******….….../

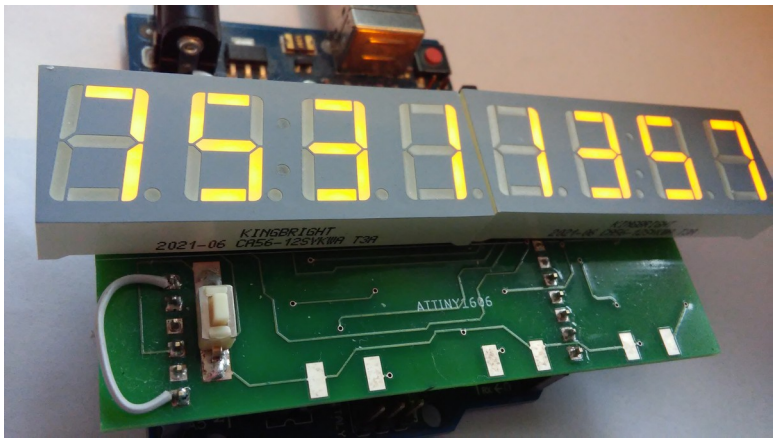
Remove the forward slash from the beginning of the line:

/****************……...Programmer code ends here**************……../

Comment out the lines

#include "Resources_UPDI_programmer.h" and

} //Only required if programmer code is included.



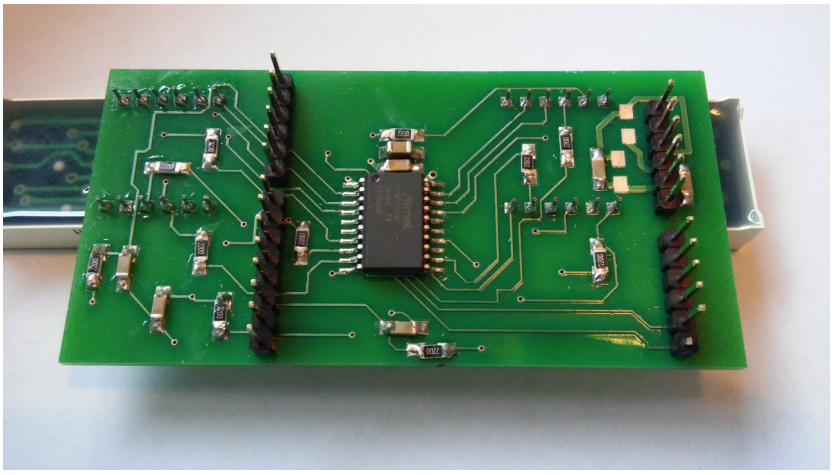
View of the display plugged into a UNO.

Note the addition of a wire link

Horizontal switches originally use for data input via the IO have not been loaded.

The single switch that is shown is used to toggle the intensity.

The UNO reset switch also resets the display pcb.



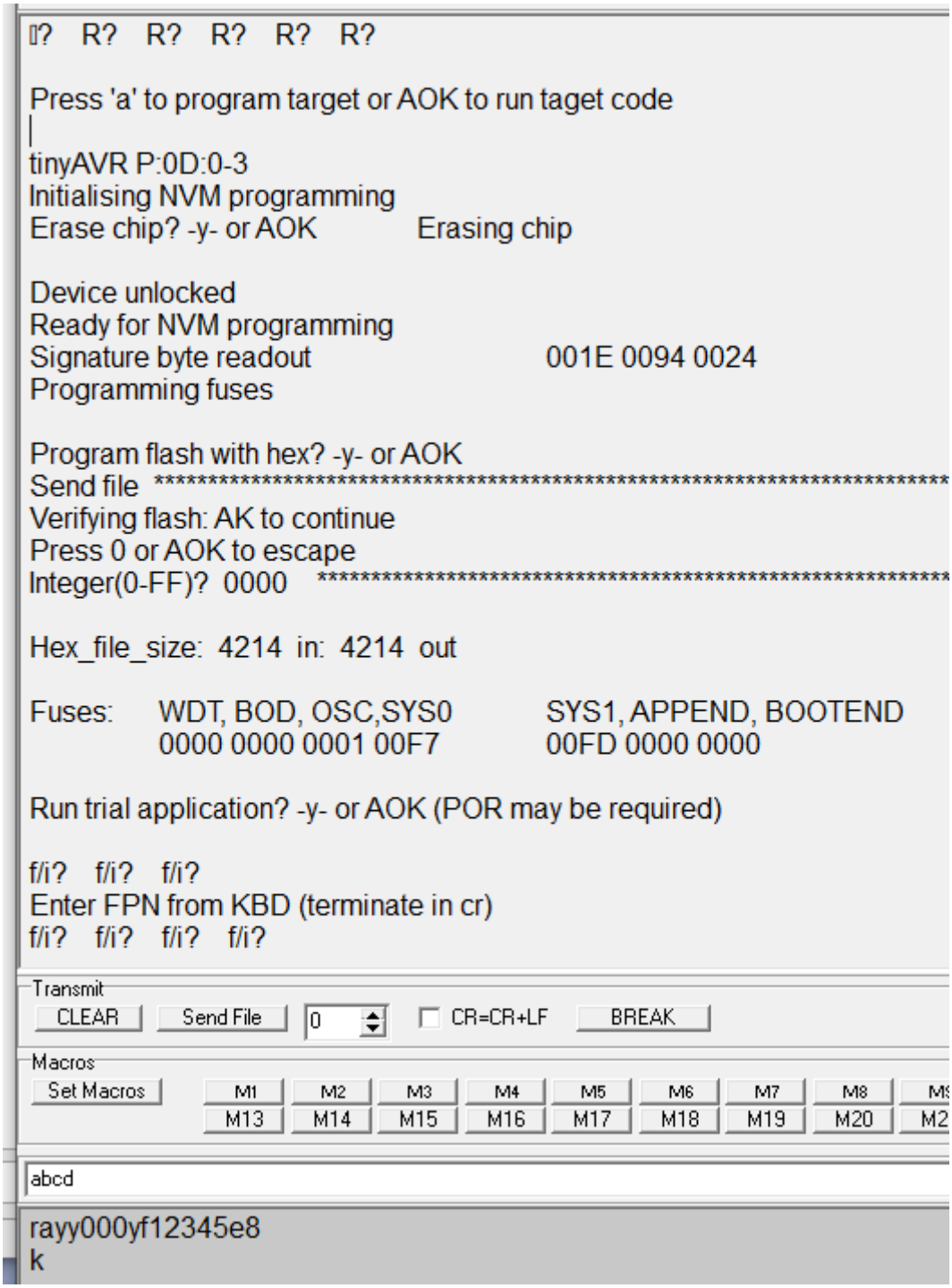
View of component side of the pcb.

Note the modified header.

This modification and the link are due to a pcb design error and were thought preferable to changing the programmer code.

Unused pads are for additional decoupling which is probably not required.

Note: Intensity can be changed after a reset and while the UNO is waiting for the user to type in a number. Otherwise it can only be changed if a global interrupt is set.



Typical terminal session

R R R R ….. is the reset user prompt
User responses are as follows:

"rayy" is used to program the 1606
"000" verifies programming which is not normally needed
"y" runs the UNO application
"f" signals that the user wants to enter a floating point number

Note: Default terminal settings are used but the baud rate is 28.8K