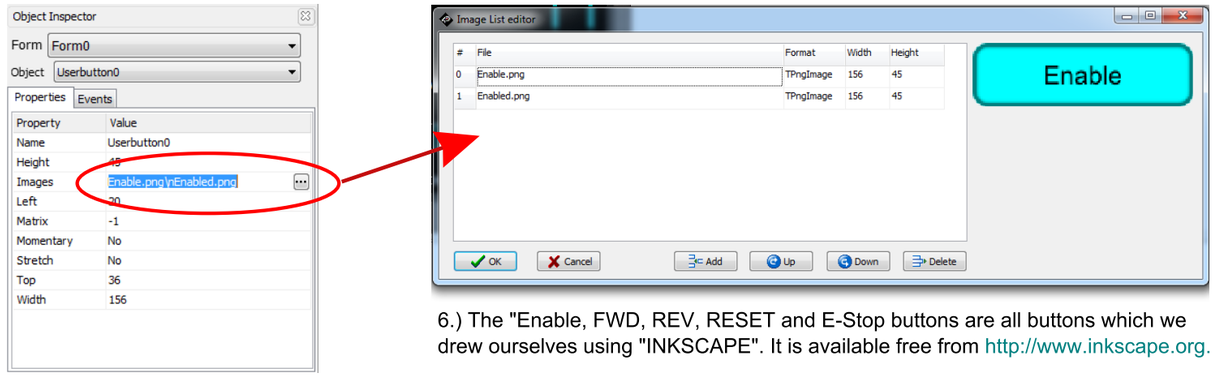
1.) This info sheet assumes that you have at least a basic familiarity with the "WORKSHOP 4" software by 4D Systems! You can download this software for free from http://www.4dsystems.com.au.

2.) 4D Systems have put a lot of effort into building up a very good Library for the Arduino IDE called "genieArduino.h" This library is intended for use with the "Visi-Genie" setup and protocol. You can find the library files at the following GITHUB repository: http://github.com/4dsystems/ViSi-Genie-Arduino-Library

3.) You should also be familiar with the following application notes by 4D systems, which can also be found at the address below: http://www.4dsystems.com.au/appnotes/ (Look for application notes 4D-AN-P4001 up to and including 4D-AN-P4024)

4.) These are all fantastic reference documents and you will be astounded, as we were, by what you can do with their display range. If you take some time and go through these, you will open up a whole new world of possibility for your Arduino projects.

5.) This page shows the Human Machine Interface (HMI screen which we put together for this tutorial. All we are showing here are the screen objects which we used to build up the HMI and what configurations are important so that you can then communicate with the HMI using your Arduino. If you are unsure of what is happening here, then READ THE APPLICATION NOTES above :)

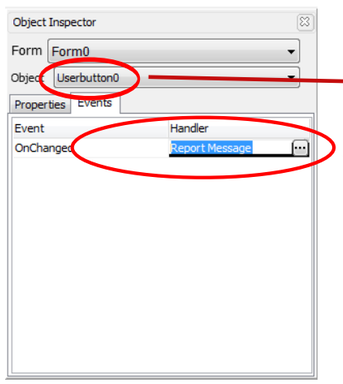


Note: The "Images" field lets you pick the button state drawings which you created.

6.) The "Enable, FWD, REV, RESET and E-Stop buttons are all buttons which we drew ourselves using "INKSCAPE". It is available free from http://www.inkscape.org.

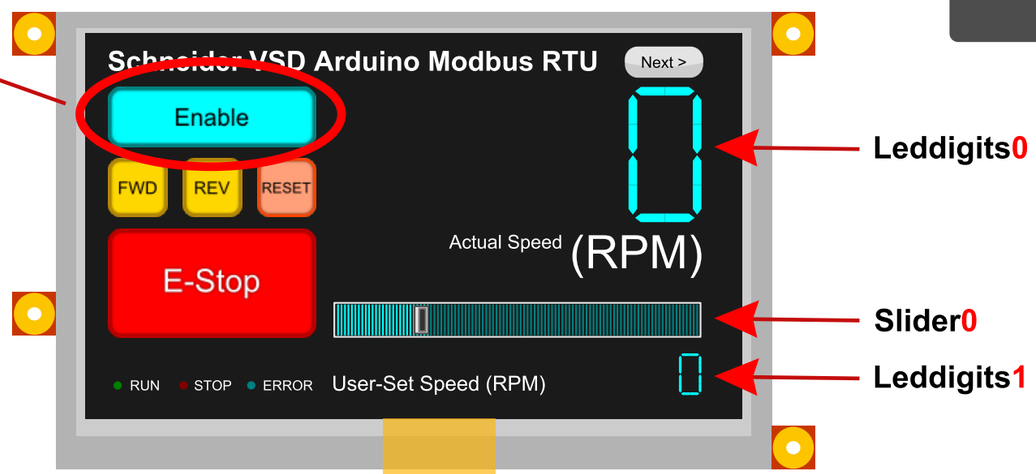
7.) You need to draw two PNG images for each button, one for the "off" state and another for the "on" state.

8.) You then use a "User Button" object to create the button and you can select the images which you drew from the "Images" field in the "Properties" tab of the "Object Inspector window in Workshop 4.



Note: If you do not set the User Button to "Report Message" then you will be banging your head against the wall wondering why you can't get the button value to come up in the Arduino code! Make absolutely sure that you have this set for each object that you hvae sending info back to your Arduino!

Userbutton0 - The "0" is the index that we use in the Arduino IDE



Leddigits0

Slider0

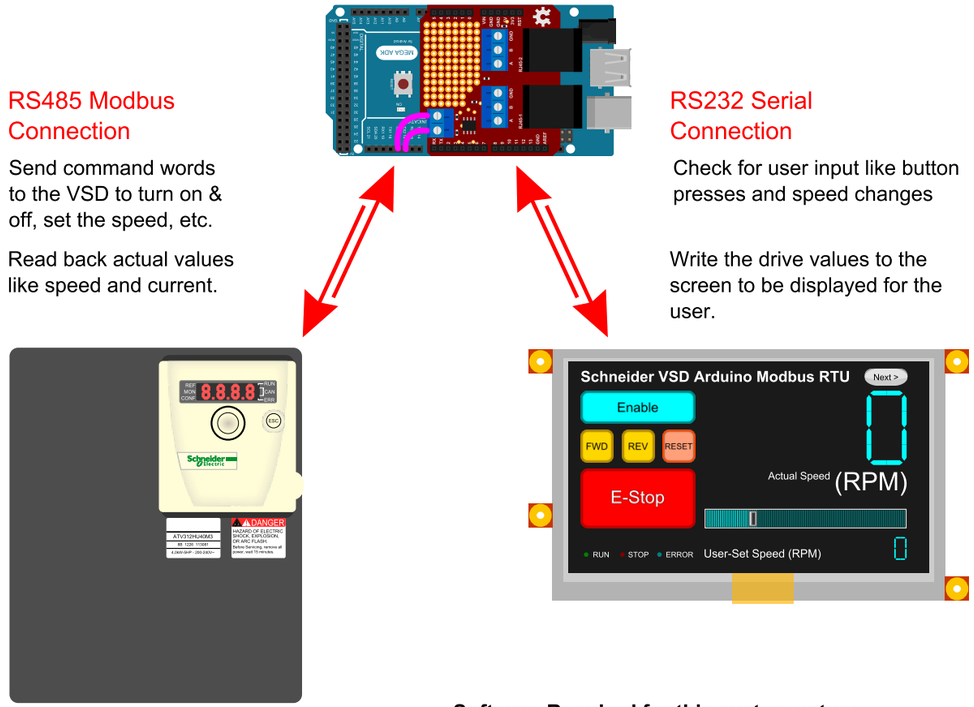
Leddigits1

9.) From the objects on the display, you will see that each is a certain "type", such as "Userbutton" or "Leddigits" or "Slider". (There are also lots of other object types too... have a look in "genieArduino.h" and the application notes from 4D Systems.)

10.) In addition to the above, each object also has an "Index" which is just a unique number for each separate object of the same "Type" on the screen.

11.) On the next sheet of this tutorial we show how to get information to and from the display using these objext "types" and "Indeces", as well as how to get information to and from the VSD using the MODBUS link which we create by using the AET RS485 shield.

Communication Overview:



Software Required for this system setup:

1. WORKSHOP4 by 4D Systems for screen design, programming and communication setup.

2. Arduino IDE with the "SimpleModbusMaterAET.h" library and the "genieArduino.h" library loaded. (see http://github.com/aetcnc/Arduino_4Dsystems_VSD)

3. Sample program from the above GITHUB repository which grabs key press information from the HMI buttons and turns them into a "BUTTONSTATE" variable so the user can tell which button has been pressed.

Schneider VSD
4D Systems uLCD-43PT
Prepared By: GHJ Date: 08/01/2014

Affordable•Engineering•Technologies

Schneider VSD + uLCD-43PT - Sheet 2/3
- Screen Basics and Reference Documents