4D SYSTEMS
TURNING TECHNOLOGY INTO ART

Arduino™ Compatible Display Module

DATASHEET

DOCUMENT DATE: **26th May 2020**
DOCUMENT REVISION: **1.9**

4Duino-2.4"

Table of Contents

4Duino – Arduino Compatible Display Module – 2.4"

4Duino – Arduino Compatible Display Module – 2.4"

4Duino – Arduino Compatible Display Module – 2.4

1. Description

The 4Duino is an Arduino™ compatible display module with built in 240x320 resolution TFT LCD Display with Resistive Touch, and Wi-Fi capabilities.

At the heart of 4Duino is an ATmega32U4 8-bit micro controller from Atmel. The same microcontroller is found on popular Arduino Leonardo. 4Duino features a 2.4" colour TFT LCD display, with resistive touch. It is powered by the feature-rich 4D Systems Picaso Graphics Processor, which offers an array of display functionality and options for any Designer/ Maker. In addition, the 4Duino features the popular ESP8266 Wi-Fi module which is pre-programmed with the AT command set firmware, enabling the 4Duino to have many Wi-Fi capabilities right out of the box.

The 4Duino also features an on-board microSD connector, and headers in the layout of an Arduino, including power pins (5V, 3.3V, GND and VIN), 20 digital IO pins, of which 7 can be used PWM outputs and 12 pins have Analog input capabilities.

The 4Duino is easily programmable with the Workshop4 IDE and its 3 different 4Duino based development environments.

The Workshop4 IDE has new functionally added, and is now able to program the on board Atmel processor using the popular Arduino programming language, just like the popular Arduino IDE, but with the added dimension of graphics – with the aid of the 4D Systems Picaso GPU. Creating Arduino based GUI's does not get any easier.

The 4Duino design has incorporated a TFT LCD Touchscreen Display and Wi-Fi solution with the intention of providing a flexible and hugely capable hardware platform. This will enable Makers and Designers to significantly improve the quality and the scope of their projects, and do so rapidly with the aid of the 4D Systems Workshop4 IDE.

The 4Duino is *the* best board to get started with electronics design, project, and education, and is perfectly suitably suited to both beginners and experts alike.

NOTE: Arduino is a trademark of Arduino Team, and all references to the word "Arduino" or use of its logo/marks are strictly in reference to the Arduino product, and how this product is compatible with the aspect of the product but is not associated with the Arduino Team in anyway.

2. Features

- ATmega32U4 with
  - 32KB Programmable Flash
  - 2.5KB Internal SRAM
  - 1KB Internal EEPROM
  - UP to 16 MIPS Throughput

- Powerful 2.4" Intelligent LCD-TFT display module powered by PICASO with
  - 14KB Programmable Flash
  - 14KB Internal SRAM
- 240 x 320 Resolution, RGB 65K true to life colours, TFT LCD Display with integrated 4-wire Resistive Touch Panel.
- ESP8266 Wi-Fi Module with
  - 802.11 b/g/n
  - Wi-Fi Direct (P2P), soft-AP
  - TCP/IP protocol stack
  - 1MB Flash for **4Duino Rev 1.2 (ESP-06)**
  - 4MB Flash for **4Duino Rev 1.3 (ESP-12S)**
- General Purpose I/O pins for user interfacing, which include
  - 20 Digital IO pins
  - of which 7 are capable of PWM
  - and 12 are capable of Analog input
- On-board USB for powering the 4Duino and programming the ATmega32U4.
- 2x5 way header for programming Picaso and ESP8266 via a 4D Systems Programming Cable or Adaptor
- On-board push-push type micro-SD memory card connector for multimedia storage and data logging purposes.
- DOS compatible file access (FAT16 format) as well as low level access to card memory.
- Display full colour images, animations, icons and video clips.
- Supports all available Windows fonts.
- 4.0V to 5.5V range operation (single supply).
- 1x 5V DC Barrel Jack for Supply.
- Module dimensions: 72.8 x 53.3 x 14.6mm. (**Rev 1.2**), 75.2 x 53.3 x 14.6mm (**Rev 1.3**).
- Weighing ~36 g.
- Display Viewing Area: 36.72 x 48.96mm.
- RoHS and CE Compliant.

4Duino – Arduino Compatible Display Module – 2.4

3. Before Getting Started

Out of the box, the 4Duino does not have the Arduino headers or ICSP header soldered. These however come in a bag and can be installed by the User if required.

Installation of these headers onto 4Duino board requires a soldering iron and solder.

Headers may not be required in cases where wires are desired to be soldered directly, or if alternate style headers are desired instead.
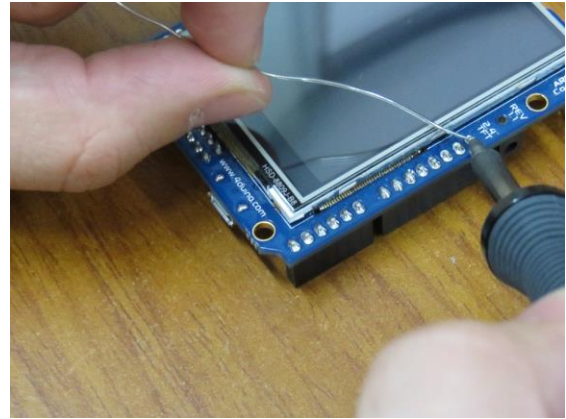
3.1. STEP1 – Insert Headers

Insert all five headers (H2, H3, H4, H5 and ICSP) in the right direction. The pins of the header should enter the bottom side (circuitry) of the board and extend out the top side (Display). Use the diagram below as reference.
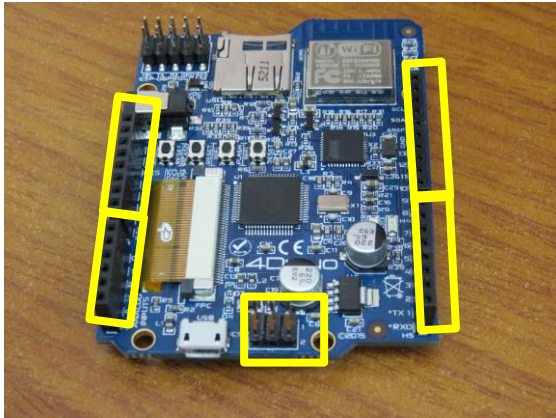
3.2. STEP2 – Before Soldering

Making sure the headers are inserted in the right orientation is essential. Then flip the board over allowing the display side to face up.

Try to align the headers perpendicular to the 4Duino PCB, otherwise shields will be difficult to attach.

3.3. STEP3 – Soldering

It is important each headers is perpendicular to the board. This will ensure any shields will slide straight onto your 4Duino without bending any pins.

Initially, solder only one pin on each header. If they are not perpendicular it will be easier to reheat that single pin, while adjusting the alignment.

After soldering one pin in each header, verify the alignment. Then, solder the remaining pins of the headers.

3.4. STEP4 – Check Solder Joints

Ensure there are no bad solder joints creating a short. If so, apply some solder wick to the joint and using your soldering iron, remove the short.

Also check for cold joints where there is not enough solder to connect the two points together.

Tip: look out for pins which are loose or joints which are not shiny. These may be due to cold joints or insufficient solder.

The photo below shows what the 4Duino will look like after the headers have been soldered in correctly.

4. Powering the 4Duino

4.1. USB Power

The 4Duino is able to be powered numerous ways, however the most common will be using the micro USB jack, which is also used to program the ATmega32U4.

The 4Duino runs off a 5.0V DC supply, and draws approximately 320mA using an average arrangement of features at any given time. This is within the suitable limits of the supply for most computer USB ports.

4.2. DC Barrel Jack

The 4Duino features a DC Barrel Jack, designed for a fine pitch DC Jack, 2.35 mm in diameter, with centre hole of 0.7mm and barrel length of 8mm or more.

A suitable 1 Amp 5.0V DC Adaptor is available from the 4D Systems website, which has been designed specifically for 4D Systems products. It features international interchangeable AC heads, to suit different AC sockets found around the world.

The Barrel Jack on board is a CUI Inc. PJ1-023-SMT if more information is required.

4.3. Powering from the Headers

The 4Duino is able to be powered off the power header (H3). There is a V_{IN} pin located at the bottom of the H3 header which accepts only 5.0V_{DC}, with a maximum input of approximately 5.5V_{DC}. Input outside this range could damage the module as some peripherals run directly off the 5V bus.

There is a pin labelled 5V, located on pin 5 of the H3 header. This is NOT designed for direct input, and is designed to pass power to other devices. This pin comes after the reverse polarity protection diode, so if the polarity of a 5V supply is connected backwards and connected to this pin, the module could be damaged. This pin will supply approximately V_{IN} -0.3 V, when the module is powered from VIN, USB or the DC Jack.

4.4. The Power Supplies

The 4Duino features an on-board 3.3V regulator. There are no 5.0V regulator, so power input to the 4Duino is required to be a regulated 5.0VDC supply.

The LCD Backlight is powered directly off the 5V bus, and therefore the backlight may be damaged if the display module is supplied with power outside the recommended ranges.

The 3.3V regulator is there to supply the PICASO, ESP8266 Wi-Fi module, microSD card and display logic, with clean regulated power. The regulator is a LP38692MP-3.3, which is capable out outputting up to 1Amp. A suitable 5VDC input is required to achieve full current output of the 3.3V regulator however.

When powering the display, multiple power sources can be connected if desired, such as USB and Barrel Jack, or USB and 5V V_{IN} Header input. There is a simple diode pair on board, so which ever has the highest voltage will take over supply to the display module. This protects multiple supplies from fighting each other. For example, the DC Barrel Jack to be used to power the display, and the USB can then be connected to program the 4Duino, without having to remove the DC Barrel Jack.

5. Block Diagram

6. Hardware Overview

5 way header (H1 Top) to program PICASO (Graphics controller)

Push-push Type microSD Socket

ESP8266 Wi-Fi module With PCB Antenna

5 way header (H1 Bottom) to program ESP8266 (Wi-Fi

5V DC Barrel Jack for Supply

Header (H3) with Power pins

PICASO Driving the 2.4" LCD Display

Header (H2) with GPIO

ATMega32U4 Microcontroller

Header (H4) with GPIO

Header (H5) with GPIO

Micro USB for programming or powering 4Duino

ICSP Header used to program the bootloader to ATmega32U4. Also the header supports SPI communication

2.4" colour TFT LCD display, 240 x 320 Resolution, RGB 65K true to life colours, with integrated 4-wire Resistive Touch Panel.
Main processor is the Atmel ATmega32U4, and Graphics are powered by the feature-rich 4D Systems PICASO Graphics Processor.

4Duino – Arduino Compatible Display Module – 2.4

7. 4Duino Pin Description and Summary

Numbers listed on the PCB next to the Headers, refer to the Symbol column of the following tables. These are the Digital and Analog pin references for use inside the IDE.
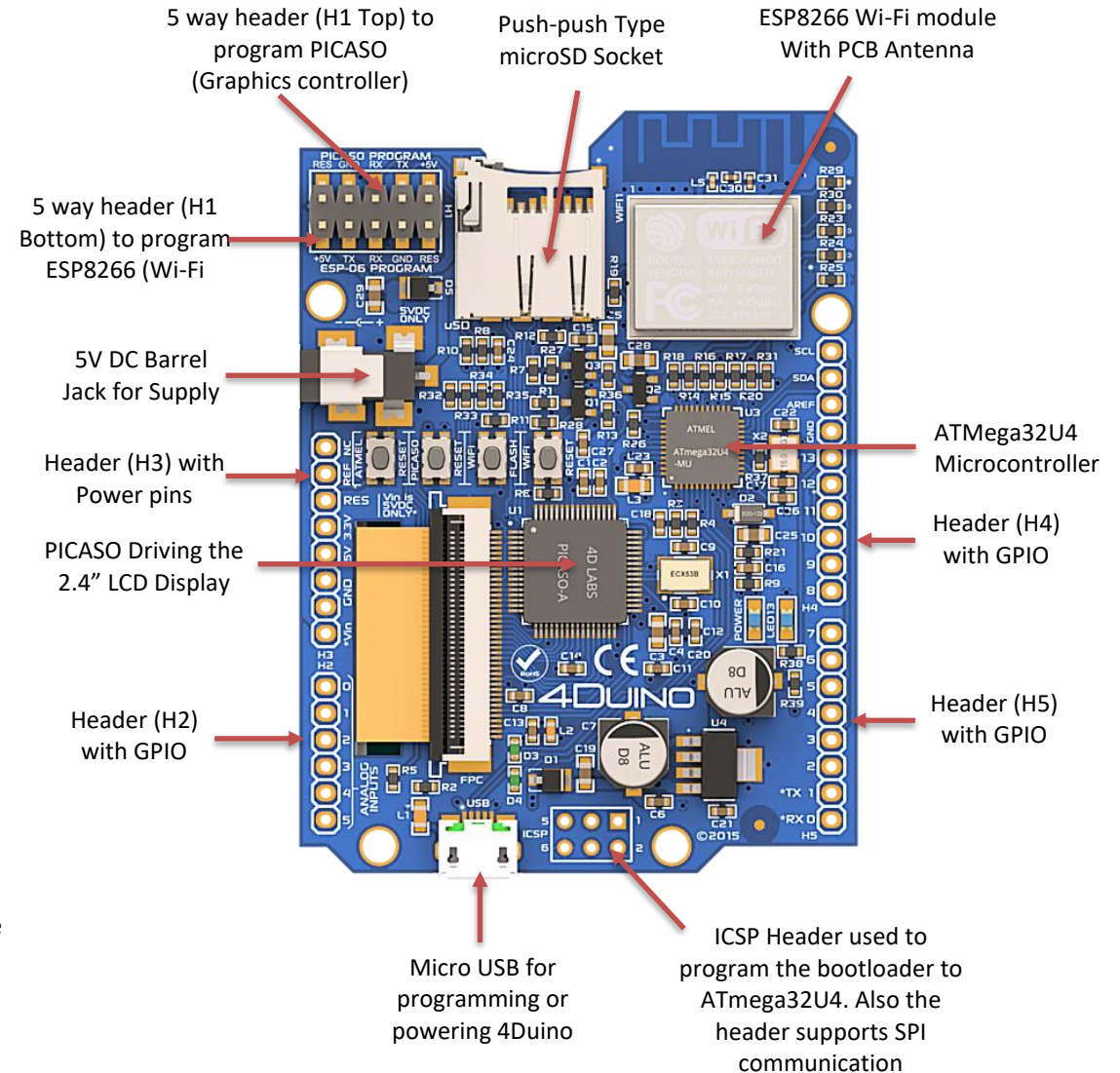Pin numbers listed in the tables below, refer to the pins used in the Schematic Diagram.

7.1. 4Duino Pin Description
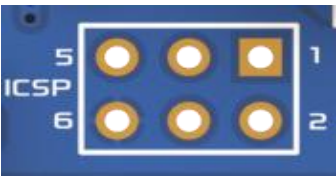
USER I/O – 30 Way FPC				
Header	Pin	Symbol	I/O	Description
H3		Refer to the Diagram in Hardware Overview Section		
	8	V$_{IN}$	P	Main Supply INPUT, 5V DC ONLY. This pin is designed to power the display module, features reverse polarity protection. Input should not exceed 5.5V DC, else damage to the Display backlight could result.
	7	GND	P	Supply Ground
	6	GND	P	Supply Ground
	5	5V	P	+5V OUTPUT pin. This pin is designed to power external devices such as Shields, not to power this module as it comes after the reverse polarity protection. Output will be approximately 4.7V with a 5.0V input on VIN
	4	3.3V	P	+3.3V OUTPUT pin, which is connected to the LP38692MP-3.3 regulator, and capable of supplying up to 1Amp. Note this 3.3V supply is also used for the Display, Wi-Fi module and microSD card.
	3	RES	I	Master Reset signal. Internally pulled up to 5V via a 10K resistor. An active Low pulse greater than 2 micro-seconds will reset the module
	2	REF	P	3.3V voltage reference with which the ATmega32U4 operates. A properly configured shield can read the REF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V
	1	NC	-	Not connected
H2		Refer to the Diagram in Hardware Overview Section		
	6	A0	I/O/A	General Purpose I/O pin with Analog Capability.
	5	A1	I/O/A	General Purpose I/O pin with Analog Capability.
	4	A2	I/O/A	General Purpose I/O pin with Analog Capability.
	3	A3	I/O/A	General Purpose I/O pin with Analog Capability.
	2	A4	I/O/A	General Purpose I/O pin with Analog Capability.
	1	A5	I/O/A	General Purpose I/O pin with Analog Capability.

I = Input, O = Output, P = Power, A= Analog Input, PWM = Pulse width modulation Output.

4Duino – Arduino Compatible Display Module – 2.4

USER I/O – 30 Way FPC				
Header	Pin	Symbol	I/O	Description
H5				Refer to the Diagram in Hardware Overview Section
	8	D0	I/O	General Purpose Input/Output. **Special:** RX Serial Pin/ External Interrupt (interrupt 2) **Note:** It is connected to PICASO for (Hardware) Serial communication.
	7	D1	I/O	General Purpose Input/Output. **Special:** TX Serial Pin/ External Interrupt (interrupt 3) **Note:** It is connected to PICASO for (Hardware) Serial communication.
	6	D2	I/O	General Purpose Input/Output. **Special:** SDA I^{2}C Pin/ External Interrupt (interrupt 1)
	5	D3	I/O/PWM	General Purpose Input/Output with PWM output capability. **Special:** SCL I^{2}C Pin / External Interrupt (interrupt 0)
	4	D4	I/O/A	General Purpose I/O pin with Analog Capability.
	3	D5	I/O/PWM	General Purpose Input/Output.
	2	D6	I/O/A/PWM	General Purpose I/O pin with Analog input and PWM output Capabilities.
	1	D7	I/O	General Purpose Input/Output. **Special:** External Interrupt (interrupt 4)
H4				Refer to the Diagram in Hardware Overview Section
	10	D8	I/O/A	General Purpose I/O pin with Analog Capability. Supports Software Serial. **Note:** It is connected to ESP8266 for (Software) Serial communication.
	9	D9	I/O/A/PWM	General Purpose I/O pin with Analog input and PWM output Capabilities. **Note:** It is connected to ESP8266 for (Software) Serial communication.
	8	D10	I/O/A/PWM	General Purpose I/O pin with Analog input and PWM output Capabilities. Supports Software Serial.
	7	D11	I/O/PWM	General Purpose Input/Output with PWM output capability. Supports Software Serial.
	6	D12	I/O/A	General Purpose I/O pin with Analog Capability.
	5	D13	I/O/PWM	General Purpose Input/Output with PWM output capability. **Special:** Built in LED connected to this pin.
	4	GND	P	Supply Ground
	3	AREF	I	Reference Voltage for analog inputs.
	2	SDA	I/O	Same pin as Digital I/O pin 2 (D2)
	1	SCL	I/O	Same pin as Digital I/O pin 3 (D3)

I = Input, O = Output, P = Power, A= Analog Input, PWM = Pulse width modulation Output.

4Duino – Arduino Compatible Display Module – 2.4

2x5 Way Programming Header Pinout (Top Left)

Header	Pin	Symbol	I/O	Description
H1	 Refer to the Diagram in Hardware Overview Section			
H1 Top	PICASO Firmware Programming (Top)	+5V	P	+5V Output Supply
		GND	P	Supply Ground
		RX	I	Asynchronous Serial Receive pin. Connect this pin to the Transmit (Tx) signal from 4D systems uUSB PA5 programming adaptor.
		TX	O	Asynchronous Serial Transmit pin. Connect this pin to the Receive (Rx) signal 4D systems uUSB PA5 programming adaptor.
		RES	I	Reset Pin
H1 Bottom	ESP8266 Firmware Programming (Bottom)	+5V	P	+5V Output Supply
		GND	P	Supply Ground
		RX	I	Asynchronous Serial Receive pin. Connect this pin to the Transmit (Tx) signal from 4D systems uUSB PA5 programming adaptor.
		TX	O	Asynchronous Serial Transmit pin. Connect this pin to the Receive (Rx) signal 4D systems uUSB PA5 programming adaptor.
		RES	I	Reset Pin

ICSP Header (Bottom middle)

Header	Pin	Symbol	I/O	Description
ICSP	 Refer to the Diagram in Hardware Overview Section			
	1	MISO	I	Master In Slave Out
	2	5V	I/O	+5V Output Supply
	3	SCK	I	Clock
	4	MOSI	O	Master Out Slave In
	5	RESET	I	Master Reset Signal. Internally pulled up to 5V via 10K resistor.
	6	GND	P	Supply Ground

I = Input, O = Output, P = Power, A= Analog Input, PWM = Pulse width modulation Output.

The 4Duino also has 4 buttons. The functionality of those buttons are described in the table below.

ICSP Header (Bottom middle)	
Button	**Description**
Refer to the Diagram in <u>Hardware Overview</u> Section	
Atmel Reset	Resets the ATmega32U4 microcontroller
PICASO Reset	Resets the PICASO graphic controller
Wi-Fi Flash	Used along with Wi-Fi Reset button to update/change firmware on ESP8266
Wi-Fi Reset	Resets the ESP8266 Wi-Fi module

The 3 Reset buttons simply reset the device in question, either the Atmel processor, the Picaso Processor, or the ESP8266 WiFi Processor. The Flash button, when held down and the reset button pressed, puts the ESP8266 WiFi Processor into its bootloader mode. This is required when flashing the ESP8266 with updated or alternative firmware.

7.2. 4Duino Pins Summary

Each of 20 digital I/O pins on 4Duino (including Analog pins) can be used as an input or output.
In addition, some pins have specialized functions:

- **Hardware Serial:** D0 (RX) and D1 (TX). Used to receive (RX) and transmit (TX) TTL serial data using the ATmega32U4 hardware serial capability. [1]
- **Software Serial:** Not all pins support Software Serial. D8, D9, D10, D11, MISO (ICSP- 1), SCK (ICSP - 3), MOSI (ISCP - 4) supports Software Serial. If multiple Software serial ports are used, only one can receive data at a time. [1]
- **I²C:** D2 (SDA) and D3 (SCL). Support I²C communication.
- **External Interrupts:** D3 (interrupt 0), D2 (interrupt 1), D0 (interrupt 2), D1 (interrupt 3) and D7 (interrupt 4). These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM:** Pins D3, D5, D6, D9, D10, D11, and D13 provide 8-bit PWM output.
- **SPI:** Pins on the ICSP header support SPI communication. Note that the SPI pins are not connected to any of the digital I/O pins as they are on the Arduino Uno, they are only available on the ICSP connector. This means that if you have a shield that uses SPI, but does NOT have a 6-pin ICSP connector that connects to the 4Duino's 6-pin ICSP header, the shield will not work.
- **LED:** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **Analog Inputs:** A0-A5, A6 - A11 (on digital pins 4, 6, 8, 9, 10, and 12). The 4Duino has 12 analog inputs, labelled A0 through A11, all of which can also be used as digital I/O. Pins A0-A5 appear in the header H3(left bottom); inputs A6-A11 are on digital I/O pins 4, 6, 8, 9, 10, and 12 respectively. Each analog input provides 10 bits of resolution (i.e. 1024 different values). By default, the analog inputs measure from ground to 5 volts.

  Note: The Headers for H2, H3, H4, H5 and ICSP will not be soldered on the board when purchased as mentioned in <u>Before Getting Started</u>.

[1]. **D0** (RX), **D1** (TX) are already occupied in order to communicate to PICASO via Hardware Serial. Hence, those pins are not available if communication between PICASO and ATmega32U4 is required. Similarly, **D8** and **D9** are occupied in order to communicate to ESP8266 via Software Serial. As a result, those pins are not available if communication between Wi-Fi module and ATmega32u4 is required. The ESP8266 could be held in reset to utilise these pins for other purposes.

8. Module Features

Some of the main features of the module are listed below.

8.1. ATmega32U4 Microcontroller

At the heart of 4Duino is an ATmega32U4 8bit micro controller from Atmel.

The same microcontroller is used in the popular official Arduino Leonardo.

The microcontroller has built-in USB communication, which eliminates the need of a secondary processor to enable USB communications.

The data sheet for the ATMega32U4 processor is available from the http://www.atmel.com website.

8.2. PICASO Processor

The TFT-LCD display is powered by the PICASO Graphics Processor from 4D-Labs.

The PICASO processor is a custom graphics processor, where all functionality including the high level commands are built into the chip.

All of the data and control signals are provided by the chip to interface directly to the display.

Powerful graphics, text, image, animation and countless more features are built right inside the chip.

The data sheet for the processor is available from the http://www.4dsystems.com.au website:
"PICASO Processor Datasheet"

8.3. ESP8266 Wi-Fi Module

The ESP8622 Wi-Fi module enables the 4Duino to have Wi-Fi capabilities.

The ESP8266 Wi-Fi SoC is a leading platform for Wi-Fi related projects or Internet of Things (IoT).

The ESP8266 supports APSD for VoIP applications and Bluetooth co-existence interfaces, it contains a self-calibrated RF allowing it to work under all operating conditions.

On the 4Duino the ATmega32U4 is offloading the Wi-Fi networking functions to the ESP8266, however it is capable of hosting an application by itself as it has a powerful processor and storage capability.

There is an almost limitless fountain of information available for the ESP8266, all of which has been provided by amazing community support.
By default, the 4Duino comes with the ESP8266 module loaded with the AT Command set, so no programming of the WiFi modules firmware is required to get it running.

Note: The Rev 1.2 uses ESP-06, and the Rev 1.3 uses ESP-12S but both module uses the same ESP8266 Wi-Fi SoC

8.4. SD/SDHC Memory Cards

The module supports micro-SD memory cards via the on-board push-push type micro-SD connector. The memory card is used for all multimedia file retrieval such as images, animations and movie clips. The memory card can also be used as general purpose storage for data logging applications. Support is available for off-the-shelf micro-SD (< 4GB) and high capacity HC memory cards (4GB and above). Memory cards up to 32GB is size can be used, however it must be noted that only a portion of this can be utilised by the FAT16 file system. See section below for more details.

Note: A microSD card capable of SPI is a requirement for all 4D Systems' display modules powered by Goldelox, Picaso or Diablo16 Processors. If a non-SPI compatible card is used, it will simply fail to mount, or may cause intermittent issues resulting in lock ups and crashing of the application. Please refer to the 4D Systems website for microSD cards offered by 4D Systems.

8.5. FAT16

All 4D Systems display modules featuring 4D Labs processors use off-the-shelf standard SDHC/SD/micro-SD memory cards (SPI Compatible Only) with up to 4GB capacity usable with FAT16 formatting.

For any FAT file related operations, before the memory card can be used, it must first be formatted correctly. Built into Workshop4 is a tool created by 4D Systems called RMPET (please refer to the Tools menu, in any Environment, inside the Workshop4 IDE).

RMPET allows the User to easily partition and format microSD cards, to make their file system ready to be used with 4D Systems modules. The formatting of the card can be done on any PC system with a card reader.

A Max of 4GB can be utilised by the FAT16 file system. The FAT partition is always first (if it exists). Any space larger than 4GB will be RAW, and can still be utilised by your 4D Systems module, using different functions.

9. Display/Module Precautions

- Avoid having to display the same image/object on the screen for lengthy periods of time. This can cause a burn-in which is a common problem with all types of display technologies, however typically only in extreme cases with LCD technology. Blank the screen after a while in order to prolong the life of the LCD.

- Moisture on the surface of a powered display should not cause any problems, however if water is to enter the display either from the front or from the rear, or come in contact with the PCB, damage will certainly occur. Wipe off any moisture gently or let the display dry before usage. If using this display module in an environment where it can get wet, ensure an appropriate enclosure is used.

- Dirt from fingerprint oil and fat can easily stain the surface of the display. Gently wipe off any stains with a soft lint-free cloth.

- The performance of the display will degrade under high temperature and humidity. Avoid such conditions when storing.

- Do not tamper with the display flex cable that is connected to the control board. This may affect the connection between the display and the driving circuitry and cause failure.

- Displays are susceptible to mechanical shock and any force exerted on the module may result in deformed zebra stripes, a cracked display cell and broken backlight

- Display modules have a finite life, which is typically dictated by the display itself. The backlight contains LED's, which fade over time. In the Specifications Section is a figure for the typical life of the display, and the criteria for which this life is dictated.

- Resistive Touch model features a touch sensitive film over the display which is sensitive to pressure. Take note when mounting the display module in an enclosure that pressure is not applied on the display, or false touches will occur, or the touch will simply not function at all.

4Duino – Arduino Compatible Display Module – 2.4

10. Hardware Tools

The following hardware tools are required for programming the 4Duino.

10.1. microUSB Cable / Drivers

A microUSB cable is required to program the 4Duino's Atmel processor. Any standard microUSB cable should be compatible, provided it is not one specifically designed for charging only. This cable can also be used to power the 4Duino.
Note: A microUSB cable is not sold with the 4Duino.

4D Systems has its own signed Microsoft driver for the 4Duino, which is available for download from the 4Duino product page on the 4D Systems website.

No driver is required for Mac or Linux, if those operating systems are used. Note though, Workshop4 is only available for Windows, so using the 4Duino with Mac or Linux will mean only the Arduino IDE can be used to program the 4Duino (unless via a Virtual Machine).

10.2. 4D Programming Adaptor

The uUSB-PA5 Programming Adaptor is an optional addition to the 4Duino, and is not required for out-of-the-box usage. It however provides a number of benefits to the User, as it allows the firmware on the ESP8266 to be updated or changed, and also allows the 4D Systems Picaso Processors firmware (formally known as a PmmC) to also be changed. These may be required for bug fixes or feature enhancements, but is required in order to use the Genie environment, which provides Drag/Drop graphics building. This programming Adaptor is available by purchasing a Starter Kit for the 4Duino, or it can be purchased separately. It is the standard programming adaptor used by 4D Systems for other 4D Systems products. The 4D Programming Cable can also be used.

The uUSB-PA5 Programming Adaptor are available from 4D Systems, www.4dsystems.com.au

Using a non-4D programming interface could damage your Picaso processor, and **void your Warranty.**
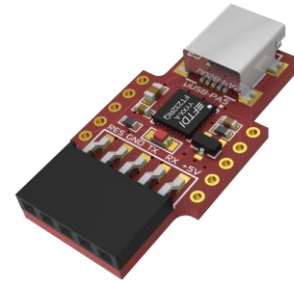
uUSB-PA5 Programming Adaptor

11. Programming the 4Duino

There are 2 IDE's available to program the 4Duino. Using the Arduino IDE, or using the Workshop4 IDE.

Using the Workshop4 IDE provides additional graphical benefits over using the Arduino IDE, however the Worskhop4 IDE is only Windows based (unless via a Virtual Machine).

If using the Workshop4 IDE, the 4Duino board will be added as part of the Workshop install so it is able to be programmed in the Arduino IDE also. If the Workshop4 IDE is not going to be used at all, then the 4Duino board needs to be added into the Arduino IDE using the Arduino Board Manager. More details below.

11.1. Arduino IDE

4Duino is 100% Arduino IDE compatible. The 4Duino can be directly programmed via Arduino IDE like any other Arduino module.

The 4Duino must be added to the Arduino IDE using the Boards Manager, after the 4D Systems JSON file has been entered into the Arduino IDE preferences.

Simply add the following JSON URL into the preferences window of the Arduino IDE, found in the File Menu, and add the JSON URL in under the Additional Boards Manager URLs field:

https://github.com/4dsystems/4Duino/raw/master/package_4dsystems_index.json

You can then simply open up the Board Manager from the Tools menu, at the top of the Board selection.

Simply click on the listing for 4D Systems AVR Boards by 4D Systems, select the version that is less than or equal to your version of the Arduino IDE (At the time of writing, 1.6.7 and 1.6.10 were available) and click the Install button. Once complete, the 4D Systems

section should be available in the Boards listings, and the 4Duino - 2.4" should be available to select.

Note: Arduino IDE 1.6.10 has known issues on some PC's, preventing downloading. 1.6.9 might be a better choice until Arduino releases a fix.

Alternatively, the Workshop4 IDE can be used to program the 4Duino, just as the Arduino IDE does.

This is possible due to the integration of the Arduino compiler via the Arduino IDE, which allows Arduino sketches to be written and compiled from within the Workshop4 IDE, which then provides WS4 the benefit of adding graphical widgets and features to the 4Duino, which would otherwise not be available when using the Arduino IDE.

When you use the Arduino IDE by itself, with no intention of using the Workshop4 IDE, you are limited to programming the Atmel processor via the Arduino IDE only, and interfacing to the Picaso Processor using 4D Systems predefined Serial Command Set, and interfacing to the ESP WiFi module using the preloaded AT Command set.

If you use the Workshop4 IDE, you can program the Atmel Processor and the Picaso Graphics Processor, if you like. You also have the opportunity to create graphics which you can call using Arduino code, or you can program the Picaso to display the graphics itself. Many options are possible.
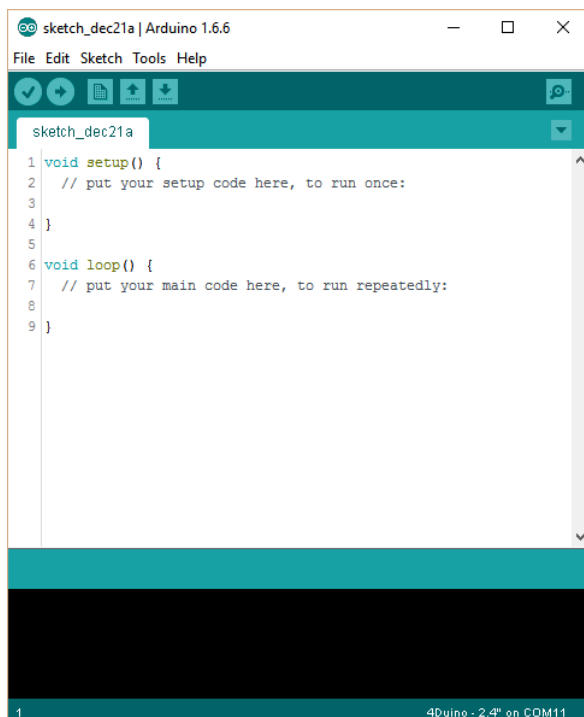
11.2. 4D Systems – Workshop4

Workshop 4 is a comprehensive software IDE that provides an integrated software development platform for all 4D Systems Intelligent Display Modules, including the 4Duino.

Workshop4 allows rapid development of applications, and on the 4Duino can take full advantage of the powerful Picaso Graphics Processor along with the Atmel ATmega32U4.

The Workshop IDE can create/edit Arduino code and verify/compile and then load the code into the 4Duino's Atmel processor, without having to use the Arduino IDE directly. It makes it possible to create both simple and complex graphical user interfaces which take into account both the Atmel and the Picaso processors, as if they were one and the same.

Note: Workshop requires the Arduino IDE to be installed as Workshop calls the Arduino IDE for compiling the Arduino sketches. The Arduino IDE however is not required to be opened or modified to program the 4Duino, once it has been set up. Please follow the steps found in the previous section to install and add the 4Duino to the Arduino IDE, prior to installing the Workshop4 IDE.

When Workshop4 is started, it presents the User with a screen to start a new project, or to load a project. Upon selecting to start a new project, another screen is displayed, presenting the wide range of 4D Systems products available to be programmed or configured by the Workshop4 IDE, of which one is the 4Duino.

The Workshop 4 IDE supports multiple development environments for the User, to cater for different user requirements and skill levels.

In the case of the 4Duino, there are 3 development Environments available.

- The **4D Arduino - Basic Graphics** environment enables the user to write Arduino code in its natural form to program 4Duino. It requires no uSD card and allows graphics primitives to be dragged and dropped on the screen and placed in your code. It utilises the Picaso Serial library, and therefore embraces the full set of Picaso Serial SPE Commands are available to the User, to produce the Graphical User Interface required.

- A visual programming experience, suitably called **4D Arduino - Extended Graphics**, enables drag-and-drop type placement of Workshop4 objects to assist with Arduino code generation and allows the user to visualise how the display will look while being developed. A uSD card will be required in order to hold the graphics. It utilises the Picaso Serial library, and therefore embraces the full set of Picaso Serial Commands is available to the User, to produce the Graphical User Interface required.

- An advanced environment called **4D Arduino - Genie Graphics** utilises the popular ViSi-Genie technology, which is available on other 4D Systems Intelligent Displays. It utilises the genieArduino library, and allows rapid development of a GUI due to the powerful tools and widgets Workshop4 provides. Simply lay the display out with the objects you want, set the events to drive them and then utilise the genie functions inside the Arduino sketch to read or write information to those objects from the Atmel processor. This is a very clean and rapid development environment. **Coming Soon.**

The Workshop 4 IDE is available from the 4D Systems website. www.4dsystems.com.au

For comprehensive manuals on the Workshop 4 IDE Software, the language, and its environments, refer to the documentation from the 4D Systems website, on the Workshop 4 product page.

11.3. 4D Arduino - Basic Graphics

The 4Duino's Basic Graphics environment is used to write Arduino code in its natural form, allowing the User to take advantage of the features of the Atmel processor and the Picaso processor, by utilising the Picaso Serial Library and the hundreds of graphics commands available.

The Basic Graphics environment provides the user with a simple yet effective programming environment allowing the usage of Serial commands where pure Arduino code can be written, compiled and loaded into the 4Duino.
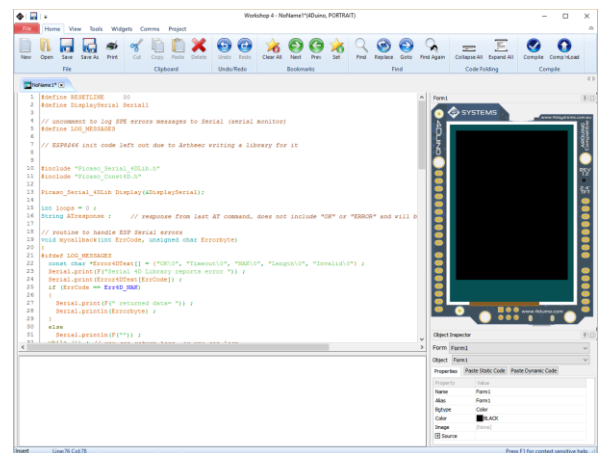
4D Arduino - Basic Graphics is designed primarily for use when no uSD card will be used with the display. Graphics primitives can be dragged and dropped onto the screen and the place in your code where the primitive is to be used can be located and the code pasted into the editor.

4D Arduino - Basic Graphics is a very powerful environment, for those use to developing with minimal GUI aid, or for those developing complex systems where no aid is required. It is also perfect for education, where graphic concepts can be learnt and studied, and exercises created to provide students with various graphical challenges to learn and conquer.

The 4D Arduino - Basic Graphics environment is the perfect starting environment for anyone new to 4D Systems modules, but also provides the greatest flexibility as there are no restrictions as to what can be coded.

4Duino – Arduino Compatible Display Module – 2.4

4Duino – Arduino Compatible Display Module – 2.4

11.4. 4D Arduino - Extended Graphics

4D Arduino - Extended Graphics was designed to make the creation of graphical displays a more visual experience while allowing the usage of powerful Picaso Serial command set still.

4D Arduino - Extended Graphics is a great software tool that allows the user to see visually what their desired graphical layout will look like, before the project is compiled and downloaded. This can make the creation of user interfaces a nicer experience, as when writing code only, the User can only see what they have created once it has been downloaded.

Like the 4Duino Basic Gra   phics environment, there are a range of primitive widgets available to the User which do not require a microSD card. On top of this however, there is a selection of in-built button, sliders, dials, gauges, meters and more that can simply be placed onto the simulated module display.

From here each object can have its properties edited, and at the click of a button all relevant Arduino code associated with that object is produced in the user program. The user can then write Arduino code around these objects, or place the object code around their Arduino code, and utilise them in the way they choose.

A microSD card is required when using 4Duino Extend Graphics environment, due to each of the extended widgets being created with images. These images need to be stored on a microSD card for the Picaso process to then utilise them.

The 4D Arduino - Extended Graphics environment is perfect for creating a wide range of applications which require graphical objects such as gauges and sliders, and where Picaso graphics coding can still take place using the Picaso Serial Commands, such as in the 4D Arduino - Basic Graphics environment.

11.5. 4D Arduino - Genie Graphics

4D Arduino - Genie Graphics is a breakthrough in the way display modules are programmed. It is an environment like no other, a code-less programming environment that provides the user with a rapid visual experience, enabling a simple GUI application to be 'written' from scratch in literally seconds.

Pick and choose the relevant objects to place on the display, much like the 4D Arduino - Extended Graphics Environment yet without having to place any of the objects code in-line with your application code.

Each object has parameters which can be set, such as colour, range, text, etc. It is then just a case of selecting the event the object is to be associated with – such as sending the value to the Atmel or directly interfacing to another object, it is that simple. In seconds you can transform a blank display into a fully animated GUI with moving sliders, animated press and release buttons, and much more.

Once the graphics have been laid out, and their Events have been set, the genieArduino library comes into play and the writing of the Atmel code begins. Using the simple yet powerful functions available, communication to and from each of the widgets placed is possible. The User doesn't need to worry about primitive graphics or graphical functions in the 4D Arduino - Genie Graphics environment, only about sending and receiving data to the widgets placed.

4D Arduino - Genie Graphics provides the user with a feature rich rapid development environment, second to none.
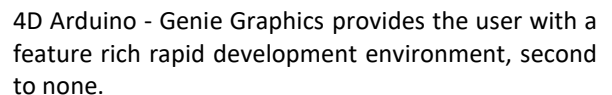
The genieArduino library is available from the 4D Systems Github repository, which can be found at this URL: https://github.com/4dsystems/ViSi-Genie-Arduino-Library

4Duino – Arduino Compatible Display Module – 2.4

Note: In order to program in Genie environment a uUSB-PA5 adapter is required. If the Genie environment is used, the User will need to download the SPE application onto PICASO in order to return the 4Duino to its default state, to use the other two environments (4Duino Basic or Extended Graphics environments) again. This is discussed in detail in the next section.

11.6. Downloading SPE to PICASO

The SPE application (Serial Platform Environment) is a program which is loaded by default into the Picaso processor, which enables it to be communicated to from the Atmel processor, using the Picaso Serial Command Set. This is a very powerful command set, basically giving full access to the 4D Graphics Language via serial commands from a Host (the Atmel).

The SPE application comes loaded by default on all 4Duino modules, however if the 4Duino has been programmed using the 4D Arduino - Genie Graphics environment, and it is desired to go back to the 4D Arduino - Basic Graphics or 4D Arduino - Extended Graphics environments, then the SPE application will need to be loaded back on to the Picaso processor. It is a simple process, but requires a uUSB-PA5 programming adaptor.
Start the Workshop4 IDE, and select the 4Duino, and select the 4D Arduino - Basic Graphics or 4D Arduino - Extended Graphics environment.

Connect your 4Duino to a uUSB-PA5 programming adaptor, connecting it to the top of the 2x5 way header, on the side marked PIACSO PROGRAM (H1 Top), taking note of the polarity and matching the signals.

Click on the Comms tab in the Workshop4 IDE and select the COM port associated with the uUSB-PA5. Click on the traffic light circle, to scan the COM port for the 4Duino module. When found, it should present itself in the status window.

Note: The status window will display a uLCD-24PTU. This is correct, as the uLCD-24PTU circuit is virtually the same as what is found on the 4Duino.

Click on the Tools tab in the Workshop4 IDE, and click on Load SPE. This will immediately program the Picaso processor via the uUSB-PA5, with the SPE application.

You should see a Splash Screen show up on the display, showing that the SPE application has been loaded. You are now free to use the 4D Arduino - Basic Graphics or 4D Arduino - Extended Graphics environments once again.

11.7. PmmC/Firmware Programming

PICASO chip level configuration is available as a PmmC (Personality-module-micro-Code) file, which can be likened to traditional Firmware.

A PmmC file contains all of the low level micro-code information (analogy of that of a soft silicon) which define the characteristics and functionality of the device. The ability of programming the device with a PmmC file provides an extremely flexible method of customising as well as upgrading it with future enhancements.

It also contains the initialisation and parameters associated with the particular display that is to be connected to the Picaso processor, along with product specific settings and parameters.

4Duino comes with the PmmC pre-installed. However, in order upgrade/change firmware follow the steps below.

Start the Workshop4 IDE, and select the 4Duino, and select any of the environments, they can all perform this procedure.

Connect your 4Duino to a uUSB-PA5 programming adaptor, connecting it to the top of the 2x5 way header, on the side marked PIACSO PROGRAM (H1 Top), taking note of the polarity and matching the signals.

Click on the Comms tab in the Workshop4 IDE and select the COM port associated with the uUSB-PA5. Click on the traffic light circle, to scan the COM port for the 4Duino module. When found, it should present itself in the status window.
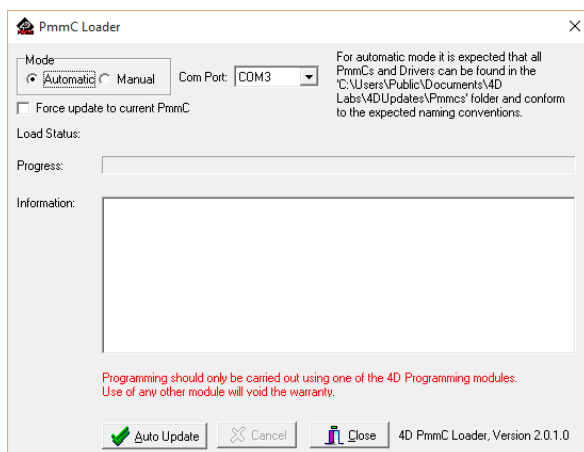
Note it will display a uLCD-24PTU. This is correct, as the uLCD-24PTU is the same as what is found built into the 4Duino.
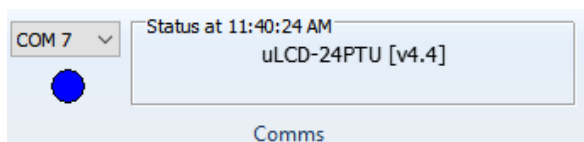
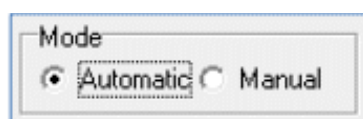Select PmmC loader from the Tools menu
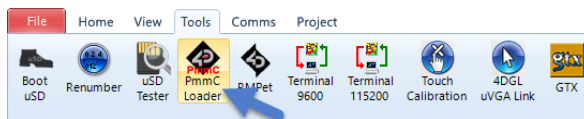
The PmmC Loader window will appear. There are 2 different ways to update the firmware.

Auto update mode:
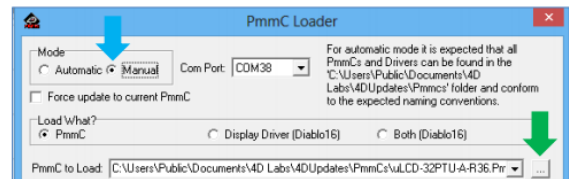By default, the mode is set to automatic.

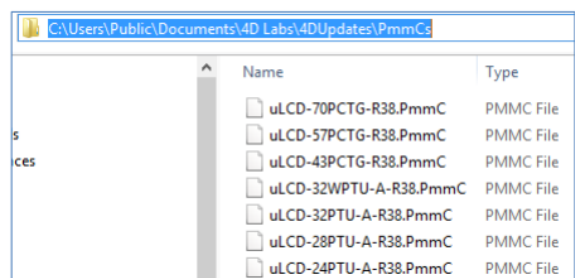Click on the Auto Update button at the bottom part of the window.
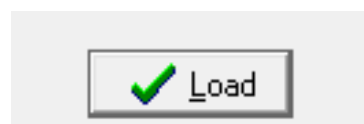

Manual update mode:
Use the manual update mode to manually select the desired PmmC file to be downloaded onto the display module.

During installation of Workshop, the latest PmmC files are copied to the default folder shown below (Windows 8).

Once the file has been selected, click the Load button to program the 4Duino with the PmmC.

If respective uploading is successful click on the Close button to exit.

NOTE: Do not attempt to use a standard FTDI programmer or other USB to Serial programmer. Use a 4D Systems Programmer, such as a uUSB-PA5. 4D Systems processors require a specific reset sequence to program correctly, and using a 3rd party programmer could result in a bricked processor.

Using a non-4D programming interface to load the PmmC onto the Picaso Processor could damage your module, and **void your Warranty.**

4Duino – Arduino Compatible Display Module – 2.4

11.8. ESP WiFi Firmware Programming

The ESP-12S module used on the 4Duino, comes preprogramed with the AT command set firmware. If an updated command set is desired, or if another program is wanted to be loaded on to the ESP-12S instead, then the default AT command set will be lost.

The following procedure outlines how to reload the ESP-12S with the AT command set.

Please download the AT command set files from the 4Duino product page on the 4D Systems website.

Extract the Files to a known location on your hard drive that is easy to access.

Run the application:
ESPFlashDownloadTool_v3.6.6.exe

Note: 4Duino Rev 1.2 uses ESP-06, but the procedure for firmware programming should be the same.

The application window will appear.
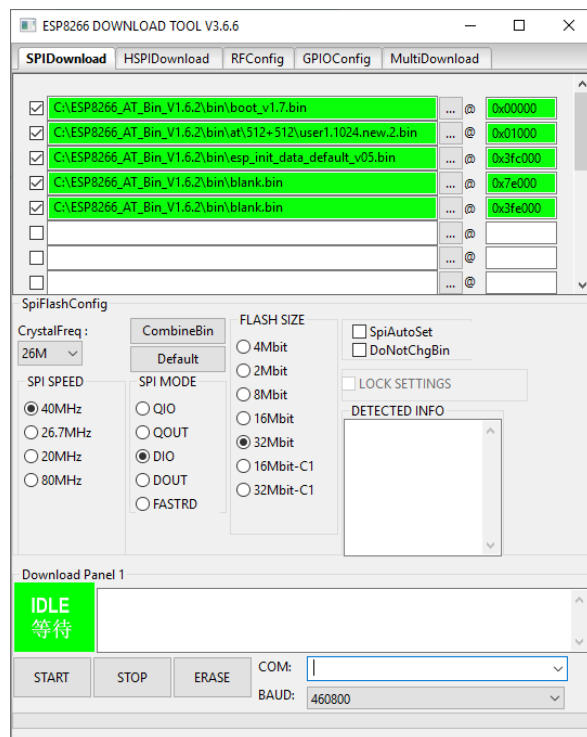
It will be noticed there are 5 different files preloaded into the Download Path Config boxes. These need to be changed to the location where the files were extracted.

Click the [...] mark and locate the path to the file it is referring to.
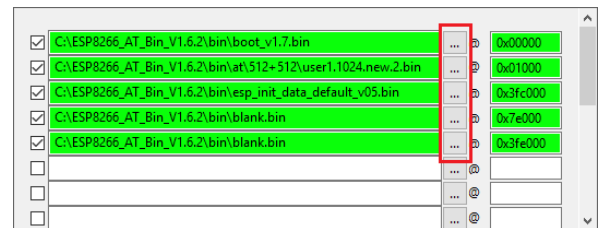
The lines will highlight if the files are found.

Ensure the Addresses are correct as per the picture, and set to the following:

boot_v1.7.bin – **ADDR 0x0000**
user1.1024.new.2.bin – **ADDR 0x01000**
esp_init_data_default.bin – **ADDR 0x3fc000**
blank.bin – **ADDR 0x7e000**
blank.bin – **ADDR 0x3fe000**

All the files are located in the root folder of that zip you just extracted, next to the ESPFlashDownloadTool_v3.6.6.exe file.

Connect the uUSB-PA5/uUSB-PA5-II programming adaptor to the 4Duino, on the ESP-12S PROGRAM side of the 2x5 way header, H1 Bottom.
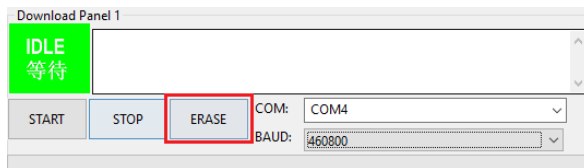

Then check the com port of the 4D Programming cable and change it accordingly in the application.
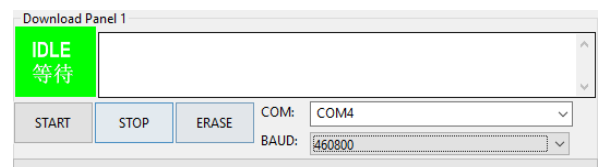

Before anything is programmed, the ESP8266 module needs to be wiped blank. It is always best to do this before programming in any new firmware, as residual data can be left which can cause unexpected results.

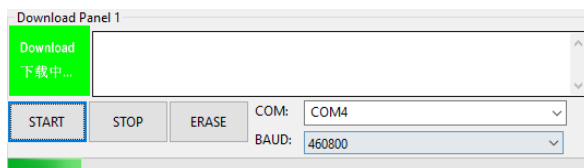To wipe the ESP-12S blank, click the 'ERASE' button

Now the ESP-12S needs to be placed into Bootloader mode. To do this, simply press and hold the WiFi Flash button on the 4Duino, and press the WiFi Reset button for a second and then release the Reset button. Keep holding the WiFi Flash button. A second or so later, you can release the WiFi Flash button. The ESP-12S should now be in Bootloader mode.
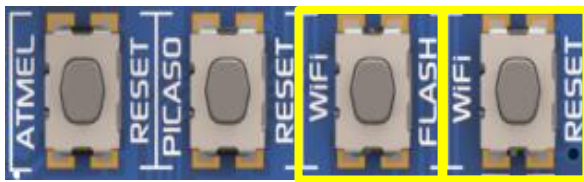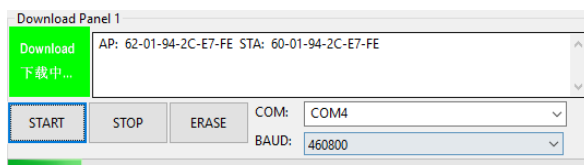

Press the Start Button, to start the programming of the ESP-12S module.
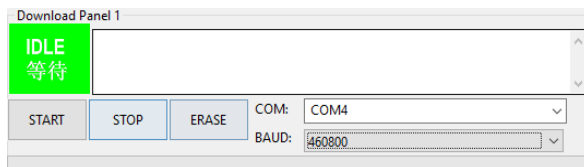
After clicking the START button the application will SYNC as shown below, and will show the MAC address of you ESP8266 shortly after.


Then the application will load and the ESP8266 is wiped blank. It will complete when the progress bar goes to the very right hand side.

If the initiation was not successful, the loading will fail. If failed, try the procedure again till it loads.

Now the Module is ready to be programmed with the firmware. This procedure is almost same as earlier.

Tick all the files except for blank1mb.bin – **ADDR 0x0000** (The only difference in the procedure compared to before)
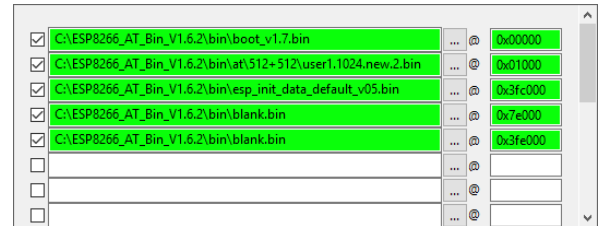
Again, the ESP-12S needs to be placed into Bootloader mode. To do this, simply press and hold the WiFi Flash button on the 4Duino, and press the WiFi Reset button for a second and then release the Reset button. Keep holding the WiFi Flash button. A second or so later, you can release the WiFi Flash button. The ESP-06 should now be in Bootloader mode.

Press the Start Button, to start the programming of the ESP-12S module. When the progress bar goes to the very right hand side, the process is complete, and the module should have been sucessfully loaded with the AT command set.

This process does not require a 4D Systems programming adaptor, however the uUSB-PA5 is compatible and able to perform this procedure.

NOTE: If there is a sketch loaded into the Atmel processor which is set to talk to the ESP module, you will either need to wipe this program by loading in a blank sketch, or you will need to hold the Atmel Reset Button throughout the entire duration you are programming the ESP module. If you do not, the programming will fail, due to the Atmel processor talking over the top of the ESP Programming and causing corruption of the data.

12. Starter Kit

4D Systems highly recommends all first time buyers of 4Duino display module, to purchase the Starter Kit when purchasing their first 4D Systems display solution.

The Starter Kit provides all the hardware that is required to explore the whole functionality of 4Duino and 4D Systems Workshop 4.

The **4D Arduino - Basic Graphics** environment can utilise every feature of the display, however, a micro-SD (uSD) card or uUSB-PA5 programming adaptor is not required.

The **4D Arduino - Extended Graphics** environment is the same as Designer in terms of feature utilisation, but is image based so requires a uSD card. A uUSB-PA5 programming adapter is not required in this case.

The **4D Arduino - Genie Graphics** environment is also image based, and therefore requires a uSD card and a uUSB-PA5 programming adapter.

Starter Kits typically include:
- 4Duino Display module
- uUSB-PA5-II Adaptor
- 4GB Industrial micro-SD Card
- 5-way cable for easy connection between the uUSB-PA5 and the 2x5 way programming header
- Quick Start Guide

Please refer to the 4D Systems website for current components included in the Starter Kit.

Simply select the Starter Kit option when purchasing the chosen display module on the 4D Systems shopping cart, or from your local distributor.

4Duino – Arduino Compatible Display Module – 2.4

13. Mechanical Details

4Duino – Arduino Compatible Display Module

	NAME	DATE
DRAWN	4D	18/06/2019
CHK'D		
APPV'D		

UNLESS OTHERWISE SPECIFIED:
DIMENSIONS ARE IN MILLIMETERS
SURFACE FINISH:
TOLERANCES:
LINEAR: 0.3mm
ANGULAR:

TITLE:
4Duino 2.4" - Arduino Compatible Intelligent Display Module

A3

DO NOT SCALE DRAWING   REVISION 1.3   SCALE: 2:1   SHEET 1 OF 1

4D SYSTEMS

14. Schematic Details

4Duino – Arduino Compatible Display Module

ARDUINO USB Interface

ARDUINO ICSP Interface / RESET

PICASO PROCESSOR

FPC TO DISPLAY

INTER-PROCESSOR INTERFACES

BACKLIGHTS

ARDUINO INTERFACES

ESP-12S ESP8266 WiFi Module

ARDUINO ATMega32U4 Processor

PROGRAMMING

DC IN + 3.3V REG

LEDS

On every boot/reset/wakeup, GPIO15 MUST keep LOW, GPIO2 MUST keep HIGH.

GPIO0 HIGH -> RUN MODE, LOW -> FLASH MODE.

Arduino programmed over USB
ESP programmed over PA5 - half of a 5x2 header. Normally AT Commands.
Picaso programmed over PA5 - other half of 5x2 header. Normally SPE Commands.

http://www.4dsystems.com.au

Title: 4Duino 2.4"
Revision: 1,3
Date: 11/12/2018
Time: 11:17:29 AM
File: 4Duino-Picaso-REV1,3.SchDoc

4D SYSTEMS
TURNING TECHNOLOGY INTO ART

15. Specifications

ABSOLUTE MAXIMUM RATINGS
Operating ambient temperature .... -20°C to +70°C
Storage temperature .... -30°C +80°C
Voltage on any digital input pin with respect to GND .... -0.3V to VCC+0.5V
Voltage on VCC with respect to GND .... -0.3V to 6.0V
Maximum current sunk/sourced by any pin 40.0mA
Maximum current sunk/sourced by all ports 200.0mA
NOTE: Stresses above those listed here may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the recommended operation listings of this specification is not implied. Exposure to maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS					
Parameter	**Conditions**	**Min**	**Typ**	**Max**	**Units**
Supply Voltage (VCC)	Stable external supply required	4.0	5.0	5.5	V
Operating Temperature		-10	--	+60	°C
Input Low Voltage (VIL)	5.0V, all pins	0	--	0.2VCC	V
Input High Voltage (VIH)	5.0V, all pins	0.7VCC	--	VCC+0.5	V
Reset Pulse	External Open Collector	2.0	--	--	µs
Operational Delay	Power-Up or External Reset	500	--	3000	ms

GLOBAL CHARACTERISTICS BASED ON OPERATING CONDITIONS					
Parameter	**Conditions**	**Min**	**Typ**	**Max**	**Units**
Supply Current (ICC)	5V Supply	--	320	--	mA
Display Endurance	Hours of operation, measured to when display is 50% original brightness	--	30000	--	H
Touch Screen Endurance	Number of touches/hits with a 12.5mm tip at a rate of 2x per second with 250gf force	--	1M	--	Touches
	Slide stylus on screen, 100gf force, 60mm/s speed with a 0.8mm polyacetal tip stylus pen	--	100K	--	Slides
Touch Screen Transparency		80	--	--	%
Touch Screen Operational Force	Only use Finger or Stylus, do not use anything sharp or metal	20	--	100	gf

LCD DISPLAY INFORMATION		
Parameter	**Conditions**	**Specification**
Display Type		TFT Transmissive LCD
Display Size		2.4" Diagonal
Display Resolution		240 x 320 (Portrait View)
Display Brightness	5V Supply – Resistive-Touch	304cd/m2
Display Contrast Ratio	Typical	250:1
Display Viewing Angles	Above Centre	60 Degrees
	Below Centre	80 Degrees
	Left of Centre	80 Degrees
	Right of Centre	80 Degrees
Display Viewing Direction		6 o'clock Display (Optimal viewing is from below when in Portrait mode)
Display Backlighting	White LED Backlighting	1x4 Parallel LED's
Pixel Pitch		0.153 x 0.153mm (Square pixels)
Pixel Density	Number of pixels in 1 row in 25.4mm	166 DPI/PPI

16. Hardware Revision History

Revision Number	Date	Description
1.2	30/12/2015	Initial Public Release Version
1.3	18/06/2019	Changing Wifi Module from ESP-06 to ESP-12S

17. Datasheet Revision History

Revision Number	Date	Description
0.1	xx/xx/2015	Internal Use Only
1.0	16/12/2015	Initial Public Release
1.1	21/12/2015	Change in install instructions for Arduino IDE JSON
1.2	03/08/2016	Final review and update of information for formal product release
1.3	08/05/2017	Updated specs, cosmetic changes
1.4	09/08/2017	Updated Drawing, fixed Page 10 mistake SCL/SDA listing backwards
1.5	19/06/2018	Updated the URL for the JSON file for Arduino IDE
1.6	04/03/2019	Cosmetic changes to 4Duino Datasheet
1.7	19/06/2019	Addition of Mechanical Details and Schematic for Hardware Revision 1.3
1.8	16/09/2019	Updated JSON file link
1.9	26/05/2020	Updated JSON file link

4Duino – Arduino Compatible Display Module – 2.4

18. Legal Notice

Proprietary Information

The information contained in this document is the property of 4D Systems Pty. Ltd. and may be the subject of patents pending or granted, and must not be copied or disclosed without prior written permission.

4D Systems endeavours to ensure that the information in this document is correct and fairly stated but does not accept liability for any error or omission. The development of 4D Systems products and services is continuous and published information may not be up to date. It is important to check the current position with 4D Systems. 4D Systems reserves the right to modify, update or makes changes to Specifications or written material without prior notice at any time.

All trademarks belong to their respective owners and are recognised and acknowledged.

Disclaimer of Warranties & Limitation of Liability

4D Systems makes no warranty, either expressed or implied with respect to any product, and specifically disclaims all other warranties, including, without limitation, warranties for merchantability, non-infringement and fitness for any particular purpose.

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications.

Images and graphics used throughout this document are for illustrative purposes only. All images and graphics used are possible to be displayed on the 4D Systems range of products, however the quality may vary.

In no event shall 4D Systems be liable to the buyer or to any third party for any indirect, incidental, special, consequential, punitive or exemplary damages (including without limitation lost profits, lost savings, or loss of business opportunity) arising out of or relating to any product or service provided or to be provided by 4D Systems, or the use or inability to use the same, even if 4D Systems has been advised of the possibility of such damages.

4D Systems products are not fault tolerant nor designed, manufactured or intended for use or resale as on line control equipment in hazardous environments requiring fail – safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines or weapons systems in which the failure of the product could lead directly to death, personal injury or severe physical or environmental damage ('High Risk Activities').   4D Systems and its suppliers specifically disclaim any expressed or implied warranty of fitness for High Risk Activities.

Use of 4D Systems' products and devices in 'High Risk Activities' and in any other application is entirely at the buyer's risk, and the buyer agrees to defend, indemnify and hold harmless 4D Systems from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any 4D Systems intellectual property rights.

19. Contact Information

For Technical Support: www.4dsystems.com.au/support

For Sales Support: sales@4dsystems.com.au

Website: www.4dsystems.com.au

Copyright 4D Systems Pty. Ltd. 2000-2019.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[4D Systems](#):
  [4Duino-24](#)   [SK-4Duino-24](#)