IoT-ALE: Discovering Tiny Snakes

IoT development without the need to compile (mostly)

John 'Warthog9' Hawley

SCaLE 17x - March 2019

Quick: MicroPython vs. CircuitPython?



MicroPython



Why is this different?

<pre>artialUpdateExample : example for Waveshare 1.54*, 2.31* and 2.9* e-Paper and the same e-papers from Dalian Good Display Inc. Created by Jean-Marc Zingg based on demo code from Good Display for GDEP0150C1. I The e-paper displays are available from: https://www.aliexpress.com/store/product/Mblesale-1-54inch-E-Ink-display-module-with-embedded-controller-200x200-Communicate-via-SPI-interface-Supports/21623 http://www.aliexpress.com/store/product/AproductSpath=2007_803Sproduct_id=35120 or https://www.aliexpress.com/store/product/AproductSpath=2007_803Sproduct_id=35120 or https://www.aliexpress.com/store/product/Orout=productSpath=2007_803Sproduct_id=35120 or https://www.aliexpress.com/store/product/Dromu.scr/index.php?topic=487007.0 Good Dispay ePaper for Arduino : https://forum.arduino.cc/index.php?topic=487007.0 Good Dispay ePaper for Arduino : https://forum.arduino.cc/index.php?topic=487007.0 Good Dispay ePaper for Arduino : D. S. G. > DB, CLK > DS, DH >> DF, OHD >> OHD >> GND , 3.3V >> 3.3V mapping suggestion from Waveshare 2.9inch e-Paper to generic ESP2066 BUSY >> DE, RST >> GF02, DC >> GF100, CS >> GF1015, CLK >> GF1014, DH >> GF1013, GND >> GND , 3.3V >> 3.3V mapping suggestion for SF32, e.g. LULIN22, see/Vxriants//pins arduino.h for your board NDTE: there are variants with different pins for SF1 ! CHECK SF1 FINS OF YOUR BOARD BUSY >> D, ST >> D, CD >> R, CLS >> GF105, CLK >> GSC103, GND >> GND , 3.3V >> 3.3V mapping suggestion for AVR, UNO, NANO etc. BUSY > 7, RST >> 0, DC >> R, CS >> 10, CLK >> 13, DIN >> 11 include GoodED21301/GoodED301301, COMP >> MSI(23), GND >> GND , 3.3V >> 3.3V mapping suggestion for AVR, UNO, NANO etc. BUSY > 7, RST >> 0, DC >> R, CS >> 10, CLK >> 13, DIN >> 11 include GoodED21301/GoodED3014, COMP >> (2.2* D/w include GoodED</pre>			0
PartialUpdateExample PartialUpdateExample : example for Waveshare 1.54°, 2.31° and 2.9° e-Paper and the same e-papers from Dalian Good Display Inc. I The e-paper displays are available from: https://www.aliexpress.com/store/product/Aholesale-1.54inch-F-Ink-display-module-with-embedded-controller-200x200-Communicate-via-SPI-interface-Supports/21623 http://www.aliexpress.com/store/product/ProductAnth-2007_3835&product_id=55120 or https://www.aliexpress.com/store/product/2001_54-inch-partial-refresh-Small-size-dot-matrix-e-paper-display/600281_32815089163.html Supporting Arduino Forum Topics: Waveshare e-paper displays with SPI: http://forum.arduino.cc/index.php?topic=487007.0 Good Dispay dPaper for Maveshare 2.9inch e-Paper to generic ESP266 BUSY > DC. RST > DA, DC > DS, CS > DB, CK > DS, DT > DT, MD > MD > MD > MD > MD > 3.3V > 3.3V mapping suggestion from Waveshare 2.9inch e-Paper to generic ESP266 BUSY > DC. RST > DA, DC > DS, CS > DB, CK > DS, DT > DF, MD > MD > MD > MD > MD > MD > 3.3V > 3.3V mapping suggestion from Waveshare 2.9inch e-Paper to generic ESP266 BUSY > DC. RST > DA, DC > DS, CS > SS (CK > DS, DT > DF, MD > MD > MD > MD > MD > MD > 3.3V > 3.3V mapping suggestion for MAVESHARE 2.9inch e-Paper to generic ESP266 BUSY > DC. RST > D, DC > T, CS > SS(S), CLK > SC K(18), DIN > OFIDI3, MD > GND, 3.3V > 3.3V mapping suggestion for MAVESHARE 2.9inch e-Raper to generic ESP266 BUSY > 2, RST > 10, DC > 17, CS > SS(S), CLK > SC K(18), DIN > MOSI(23), GND > GND, 3.3V > 3.3V mapping suggestion for MAVESHARE 2.9inch e-Raper to generic ESP266 BUSY > 2, RST > 10, DC > 17, CS > SS(S), CLK > SC K(18), DIN > MOSI(23), GND > GND, 3.3V > 3.3V mapping suggestion for MAVE, MON, NANO etc. BUSY > 7, RST > 9, DC > 8, CS > 10, CLK > 13, DIN > 11 include deception.proverbisDicl.ccpp // 1.54* bW/ sinclude -dcePEDFDSOC/scopetioSC1.cpp // 1.54* bW/ sinclude -dcePEDFDSOC/scopetioSC1.cpp // 1.54* bW/ sinclude -dcePEDFDSOC/scopetioSC1.cpp // 1.54* bW/ sinclude -dcePEDFDSOC/scopetioSC1.cpp // 1.54* bW/ sinc			
<pre>initizes/conductangles = example for marginger 1/3, first min is grapher and the same expansion for ballan bood bisplay for COEPDISC1. [] for earced by Jean-Marc Zingg based on demo code from Good Display for COEPDISC1. [] the e-paper displays are available from: thtps://www.aliexpress.com/store/product/product&path=2807 8363&product_id=35120 or https://www.aliexpress.com/store/product/product&path=2807 8363&product_id=35120 or https://www.aliexpress.com/store/product/forum.arduino.cc/index.php?topic=487007.0 Woweshare = opaper displays with SPI: http://forum.arduino.cc/index.php?topic=487007.0 Good Dispay ePaper for Arduino : https://forum.arduino.cc/index.php?topic=487007.0 Http://www.shiresels.com/store/product.com/shireself to Weeso DI min Http://www.shiresels.com/shireself to Weeso DI min Http://wweesoff</pre>	artialUpdateExample	av Inc	
<pre>treated by Jean-Ard 2.htg Daace on demo code from Good Liplay for GePrUSUL. The e-paper displays are available from: https://www.aliexpress.com/store/product/Anholesale-1-54inch-E-Ink-display-module-with-embedded-controller-200x200-Communicate-via-SPI-interface-Supports/21623 https://www.aliexpress.com/store/product/Epoth-E-Ink-display-module-with-embedded-controller-200x200-Communicate-via-SPI-interface-Supports/21623 or https://www.aliexpress.com/store/product/Epoth-E-Ink-display-module-with-embedded-controller-200x200-Communicate-via-SPI-interface-Supports/21623 or https://www.aliexpress.com/store/product/Epoth-E-Ink-display-module-with-embedded-controller-200x200-Communicate-via-SPI-interface-Supports/21623 or https://www.aliexpress.com/store/product/Epoth-E-Ink-display-Module-with-embedded-controller-200x200-Communicate-via-SPI-interface-Supports/21623 fully-interface-Support displays with SPI http://forum.arduino.cc/index.php?topic=487007.0 Good Dispay effect for Muno: https://forum.arduino.cc/index.php?topic=487007.0 Good Dispay effect for Muno: https://forum.arduino.cc/index.php?topic=487007.0 BoS' or 20, RST > 00, C > 00, C > 00, C > 00, C > 00, 3.3V > 3.3V mapping suggestion for Muno.With Hite for for Dispay for Muno: https://forum.arduino.cc/index.php?topic=48001.0 BuS' > 4, RST > 9, DC > 8, C > 10, CLK > 5K101, DL > MUN = MOSIL23, GMD > GMD, 3.3V > 3.3V mapping suggestion fo</pre>	ranciacupuarezzampre , exampre for mavesnore 1.34 , 2.31 and 2.5 enaper and the same e-papers from bactan dood bispr	T T	
<pre>/ The e-paper displays are available from: https://www.aliexpress.com/store/product/Aholesale-1-54inch-E-Ink-display-module-with-embedded-controller-200x200-Communicate-via-SPI-interface-Supports/21023 http://www.aliexpress.com/store/product/Ed021-54-inch-partial-refresh-Small-size-dot-matrix-e-paper-display/600281_32815089163.html / supporting Arduino Forum Topics: //www.aliexpress.com/store/product/forum.arduino.cc/index.php2topic=482007.0 Good Dispay ePaper for Arduino : https://forum.arduino.cc/index.php2topic=482007.0 Good Dispay ePaper for Arduino : https://forum.arduino.cc/index.php2topic=480411.0 Http://www.aliexpress.com/store/product/Spinte-Spinte-480007.0 Good Dispay ePaper for Arduino : https://forum.arduino.cc/index.php2topic=480411.0 Https://www.aliexpress.com/store/product/Spinte-Spinte-480007.0 Good Dispay ePaper for Maveshare 2.0inch e-Paper to Weeos DI mini BUSY ~ 902, RST ~ D4, DC ~ 03, CS ~ 080, CLK ~ 05, DIN ~ D7, GND ~ 0HD, 3.3V ~ 3.3V mapping suggestion from Waveshare 2.0inch e-Paper to generic ESP266 WISY ~ 0ftps T ~ 16, DC ~ 17, CS ~ 55(C), CLK ~ 05 F0104, DIN ~ 0601013, GND ~ 0HD, 3.3V ~ 3.3V mapping suggestion for ESP32, e.g. LOLIN32, see/variants//pins_arduino.h for your board WIST: there are variants with different pins for SFI ! CHECK SFI PINS 0F YOUR BOARD WISY ~ 0, RST ~ 16, DC ~ 17, CS ~ 55(C), CLK ~ 35(C), DIN ~ NOSI(23), GND ~ 0HD, 3.3V ~ 3.3V mapping suggestion for AVR, UNO, NANO etc. BUSY ~ 7, RST ~ 9, DC ~ 8, CS - 10, CLK ~ 31, DIN ~ NII ~ include defe005105/CocGe012831.cpp // 1.54* b/w einclude <cgcep0.b> // select the display class to use, only one senclude <cgcep050501.ccgcp 1.54*="" b="" w<br="">einclude <cgcep050501.ccgp 1.54*="" b="" w<br="">einclude <cgcep050501.ccgp 2.3*="" b="" w<br="">einclude <cgcep050501.ccgp 2.3*="" b="" w<br="">einclude <cgcep050501507.ccgp 2.3*="" b="" w<br="">einclude <cgcep050501.ccgp 2.4*="" b="" w<br="">einclude <cgcep050501.ccgp 2.4*="" b="" w<br="">einclude <cgcep050501.ccgp 2.4*="" b="" w<br="">einclude <cgcep050501.ccgp 2.4*="" 2.5*="" b="" cgcecues0501.ccgp="" td="" w<="" w<<=""><td>created by Jean-Marc Zingg based on demo code from Good Display for GDEP015UCI.</td><td>1</td><td></td></cgcep050501.ccgp></cgcep050501.ccgp></cgcep050501.ccgp></cgcep050501.ccgp></cgcep050501507.ccgp></cgcep050501.ccgp></cgcep050501.ccgp></cgcep050501.ccgp></cgcep050501.ccgcp></cgcep0.b></pre>	created by Jean-Marc Zingg based on demo code from Good Display for GDEP015UCI.	1	
<pre>https://www.aliexpress.com/store/product/pholesale-1-54inch-E-Ink-display-module-with-embedded-controller-200x200-Communicate-via-SPI-interface-Supports/21623 http://www.bau/ed.com/index.php?route-product/productSpath=2807_8365&product_id=35120 or https://www.aliexpress.com/store/productStore/produ</pre>	The e-paper displays are available from:		
<pre>/ http://www.bluy-icd.cow/index.php?route-product/E001-154-inch-partial-refresh-Small-size-dot-matrix-e-paper-display/600281_32815089163.html / supporting Arduino Forum Topics: //waveshare e.paper displays with SPI: http://forum.arduino.cc/index.php?topic=487007.0 / Good Dispay Paper for Arduino : https://forum.arduino.cc/index.php?topic=487007.0 // Good Dispay Paper for Arduino : https://forum.arduino.cc/index.php?topic=487007.0 // BuSira Set Set Set Set Set Set Set Set Set Set</pre>	nttps://www.aliexpress.com/store/product/Wholesale-1-54inch-E-Ink-display-module-with-embedded-controller-200x200-Commu	inicate-via-SPI-interface-S	upports/21623
<pre>/ upporting Arduino Forum Topics: /www.share e.paper displays with SPI: htp://forum.arduino.cc/index.php?topic=497007.0 Good Dispay ePaper for Arduino : https://forum.arduino.cc/index.php?topic=436411.0 / mapping suggestion from Waveshare 2.9inch e-Paper to Weeos D1 mini / BuSY > 02, BST > 04, DC > 08, CS > 06, CK >> 05, DN >> 07, GHD >> 0HD, 3.3V >> 3.3V / mapping suggestion from Waveshare 2.9inch e-Paper to generic ESP8266 BuSY >> OF104, RST >> OF102, DC >> OF100, CS >> OF1015, CLK >> OF1014, DIN >> OF013, GND >> GND, 3.3V >> 3.3V / mapping suggestion for Maveshare 2.9inch e-Paper to generic ESP8266 BuSY >> OF104, RST >> OF102, DC >> OF100, CS >> OF1015, CLK >> OF1014, DIN >> OF013, GND >> GND, 3.3V >> 3.3V / mapping suggestion for ESP32, e.g., LOLINS2, see, /variants//pins Grivour board MOT: there are variants with different pins for SP1 1 CHECK SPI PINS GF YOUR BOARD BUSY >> 4, RST >> 0, DC >> 17, CS >> SS(), CLK >> SK(10), DIN >> MOSI(23), GHD >> GND, 3.3V >> 3.3V / mapping suggestion for ARM, UNO, NANO etc. / BUSY >> 7, RST >> 0, DC >> 8, CS >> 10, CLK >> 13, DIN >> 11 / include Library, include base class, make path known mclude =</pre>	<u>attp://www.buy-lcd.com/index.php?route=product/product&path=2097_8363&product_id=35120</u> or https://www.aluexpress.com/store/product/E001-1-54-inch-partial-refresh-Small-size-dot-matrix-e-paper-display/600281	_32815089163.html	
<pre>/ mapping suggestion from Waveshare 2.9inch e-Paper to Wemos DI mini BUSY > 02, RST >> D4, DC -> D3, CS >> D8, CLK >> D5, DIN >> D7, OHD >> OHD, 3.3V >> 3.3V / mapping suggestion from Waveshare 2.9inch e-Paper to generic ESP8266 BUSY >> OF(D4, RST >> OF(D2, DC >> OF(D0, CS >> OF(D14, DIN >> OF(D13, GND >> GND, 3.3V >> 3.3V / mapping suggestion for ESP22, e.g. LOLIN22, see/Variants//pins arduine.h for your board / MDT: there are variants with different pins for SP1 (CHCK SPI PINS OF YOUR BOARD / BUSY >> 4.RC >> 17, CS >> 55(5), CLK >> 55(CLK >> 55(CLK >> 55(CLK >> 6F(D14, DIN >> MOSI(23), GND >> GND, 3.3V >> 3.3V / mapping suggestion for AVR, UNO, NANO etc. / BUSY >> 4, RST >> 10, DC >> 17, CS >> 55(5), CLK >> 13, DIN >> 11 / include tibrary, include base class, make path known include <<kcepd1>DC /> 12, See a. (J. S.4* bAve #include <<kcepd1>DC /> 13, SAV >> 13, DIN >> 11 / include tibrary, include base class, make path known include <<kcepd1>DC // 2.3* b/w #include <<kcepd1>DC // 2.5* b/w #include <<kcepd1>DC // 2.3* b/w #include <<kcepd1>DC // 2.5* b/w #include <<kcepd1>DC // 2.5* b/w #include <<kcepd1>DC // 2.5* b/w #include </kcepd1>DC // 2.5* b/w #include </kcepd1>DC // 2.5* b/w</kcepd1></kcepd1></kcepd1></kcepd1></kcepd1></kcepd1></pre>	Supporting Arduino Forum Topics: Waveshare e-paper displays with SPI: <u>http://forum.arduino.cc/index.php?topic=487007.0</u> Good Dispay ePaper for Arduino : <u>https://forum.arduino.cc/index.php?topic=486411.0</u>		
<pre>/ sapping suggestion from Waveshare 2. Joinch e-Paper to generic ESP2266 / BUSY > GPI04, RST >> GPI02, DC >> GPI00, CC >> GPI015, CLK >> GPI014, DIN >> GPI013, GND >> GND, 3.3V >> 3.3V / apping suggestion for ESP32, e.g., LOLIN32, see/variants//pins.arduino.h for your board / MOTE: there are variants with different pins for SPI ! CHECK SPI PINS OF YOUR BOARD / WOTY >> 4, RST >> 10, DC >> 11, CS >> SS(5), CLK >> SS(10), DIN >> MOSI(23), GND >> GND, 3.3V >> 3.3V / mapping suggestion for AVR, UNO, NANO etc. * BUSY >> 7, RST >> 9, DC >> 8, CS >> 10, CLK >> 13, DIN >> 11 / include library, include base class, make path known mclude = GACEPDISOCI.copp // 1.54* b/W #include = GACEPDISOCI.copp // 1.54* b/W #include = GACEPDISOCI.copp // 2.3* b/W #include = GACEPDISOCI.copp // 2.3* b/W #include = GACEPDISOCI.copp // 2.4* b/W #inclu</pre>	mapping suggestion from Waveshare 2.9inch e-Paper to Wemos D1 mini BUSY -> D2, RST -> D4, DC -> D3, CS -> D8, CLK -> D5, DIN -> D7, GMD -> GMD, 3.3V -> 3.3V		
<pre>/ mapping suggestion for ESP32, e.g. LOLIN32, see/variants//pins arduino.h for your board / WDF: three are variants with different pins for SPI L LOEK SPI PINS for FVOWE BOARD / BUSY ~> 4, RST ~> 16, DC ~> 17, CS ~> SS(5), CLK ~> SCK(18), DIN ~> MOSI(23), GND ~> GND, 3.3V ~> 3.3V / mapping suggestion for AVR, UNO, NANO etc. / BUSY ~> 7, RST ~> 9, DC ~> 8, CS ~> 10, CLK ~> 13, DIN ~> 11 / include Library, include base class, make path known include <gkepdisol gkoeppisoll.cpp=""> // 1.54* b/w frinclude <gkepdisol gkoeppisoll.cpp=""> // 1.54* b/w frinclude <gkepdisol gkoeppisoll.cpp=""> // 2.3* b/w frinclude <gkepdisol gkoeppisoll.cpp=""> // 2.3* b/w frinclude <gkepdisol gkoeppisoll.cpp=""> // 2.4* b/w frinclude <gkepolysak do="" fully="" not="" partial="" support="" update<br="">final unde <gkepdisol gkoeppisoll.cpp=""> // 1.54* b/w frinclude <gkepolysak do="" fully="" not="" partial="" support="" update<br="">final unde <gkepdisol gkoeppisoliz="" gkoeppisoll.cpp=""> // 1.54* b/w final unde <gkepolysak do="" fully="" not="" partial="" support="" update<br="">final unde <gkepisol gkoe<="" gkoeppisol="" gkoepsisol="" td=""><td>mapping suggestion from Waveshare 2.9inch e-Paper to generic ESP8266 BUSY -> GPI04, RST -> GPI02, DC -> GPI00, CS -> GPI015, CLK -> GPI014, DIN -> GPI013, GND -> GND, 3.3V -> 3.3V</td><td></td><td></td></gkepisol></gkepolysak></gkepdisol></gkepolysak></gkepdisol></gkepolysak></gkepdisol></gkepdisol></gkepdisol></gkepdisol></gkepdisol></pre>	mapping suggestion from Waveshare 2.9inch e-Paper to generic ESP8266 BUSY -> GPI04, RST -> GPI02, DC -> GPI00, CS -> GPI015, CLK -> GPI014, DIN -> GPI013, GND -> GND, 3.3V -> 3.3V		
/ apping suggestion for AVR. UMO. NAMO etc. / BUSY >> 7, RST >> 9, DC >> 8, CS >> 10, CLK >> 13, DIN -> 11 / include library, include base class, make path known include <6xEPD.h> / select the display class to use, only one // include <5xCBPD150C1/cK0EPD150C1.cpp> // 1.54* b/w // include <5xCBPD150C1/cK0EPD150C1.cpp> // 2.3* b/w // include <5xCBP045201.ccSEPD12131.cpp> // 2.3* b/w // include <5xCBP045201.ccSEPD3201.cpp> // 2.2* b/w // these display do not fully support partial update // these display do not fully support partial update	amping suggestion for ESP32, e.g. LOLIN32, see/variants//pins.arduino.h for your board MDTE: there are variants with different pins for SPI (+ CHCK SPI PINS oF YOUR BOARD BUSY → 4, RST → 16, DC → 17, CS → SS(5), CLK → SCK(18), DIN → MOSI(23), GND → GND, 3.3V → 3.3V		
/ include library, include base class, make path known include GAGEPD.h> ' select the display class to use, only one #include GAGEPDISO(1/GAGEPDISO(1,cpp> // 1.54* b/w #include GAGEPDISO(1/GAGEPDISO(1,cpp> // 1.54* b/w #include GAGEPDISO(1/GAGEPDISO(1,cpp> // 2.5* b/w #include GAGEPDISO(1/GAGEPDISO(1,cpp> // 2.2* b/w #include GAGEPDISO(1/GAGEPDISO(1,cpp> // 2.2* b/w these displays do not fully support partial update #include GAGEPDISO(1/GAGEPDISO(1,cpp> // 1.54* b/w/c 150×157 #include GaGEPDISO(1/GAGEPDISO(1,cpp> // 2.5* b/w) #include GaGEPDISO(1,cpp> // 2.5* b/w) #include GaGEPDISO(1,cpp	mapping suggestion for AVR, UNO, NANO etc. BUSY -> 7, RST -> 9, DC -> 8, CS-> 10, CLK -> 13, DIN -> 11		
/ yelact the display class to use, only one 'finclude <scopedisol 1.54'="" b="" scopedisol(.cpp)="" w<br="">'finclude <scopedisol 2.3'="" b="" scopedisol(.cpp)="" w<br="">'finclude <scopedisol 2.9'="" b="" scopedisol(.cpp)="" w<br="">'finclude <scopedisol 2.2'="" b="" scopedisol(.cpp)="" w<br="">'finclude <scopedisol(scopedisol(.cpp) 2.2'="" b="" w<br="">'finclude <scopedisol(.cpp) 2.2'="" b="" w<br="">'finclude <scopedisol(< td=""><td>include library, include base class, make path known <mark>clude <gxepd.h></gxepd.h></mark></td><td></td><td></td></scopedisol(<></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol(scopedisol(.cpp)></scopedisol></scopedisol></scopedisol></scopedisol>	include library, include base class, make path known <mark>clude <gxepd.h></gxepd.h></mark>		
ш —	select the display class to use, only one include < <pre>cxx0DF0150C1/Cxx0DF0150C1.cpp> // 1.54* b/w include <<xx0df023b1 cxx0df023b1.cpp=""> // 2.13* b/w include <<xx0df023b1 cxx0df023b1.cpp=""> // 2.9* b/w include <<xx0df04271 cxx0df04272.cpp=""> // 4.2* b/w these displays do not fully support partial update these displays do not fully support partial update</xx0df04271></xx0df023b1></xx0df023b1></pre>		
			>
		Nodo22c 20MU-031CO	in an Maulthul ISBO
		Node32s, 80MHz, 92160	0 on /dev/ttyUSB0

rst:0x1 (POWERON RESET),boot:0x13 (SPI FAST FLASH BOOT) configsip: 0, SPIWP:0xee clk drv:0x00,g drv:0x00,d drv:0x00,cs0 drv:0x00,hd drv:0x00,wp drv:0x00 de:DIO, clock div:2 pad:0x3fff0018,len:4 ad:0x3fff001c,len:4732 ad:0x40078000,len:7496 ad:0x40080400.len:5512 try 0x4008114c (388) cpu start: Pro cpu up. (389) heap init: Initializing. RAM available for dynamic allocation: (392) heap init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM (398) heap init: At 3FFC4F48 len 0001B0B8 (108 KiB): DRAM (405) heap init: At 3FFE0440 len 00003BC0 (14 KiB): D/IRAM (411) heap init: At 3FFE4350 len 0001BCB0 (111 KiB): D/IRAM (417) heap init: At 40091448 len 0000EBB8 (58 KiB): IRAM (424) cpu start: Pro cpu start user code (218) cpu start: Starting scheduler on PRO CPU. tting up LEDs tting up Buttons tting up Sensor I2C tting up BME280 tting up TSL2591 e values[0]: 2172 - 21.72C values[1]: 25929420 - 1012.86hPa values[2]: 44558 - 43.51% values[0]: 48 values[1]: 21 Good itialize the Board LED as a PWM... Success break hit <ctrl>+c then enter: breathTimer.deinit() Error: [Errno 2] ENOENT croPython v1.9.4-560-g185716514 on 2018-09-20; ESP32 module with ESP32 Type "help()" for more information. >>>

Why is this different?

- Quick, iterative, development
- . Most of the advantages of Python
- 0 to blinking LED very quick
- Mostly no need to compile anything
- Lots of default functionality, and upip (library / package management!)

Why is this possible?

- Same reason IoT is becoming ubiquitous
 - $\circ~$ MCUs & CPUs are getting more powerful, and cheaper
- ESP32 on the SensorNode cost \$5.10 to place on the board.
 - Dual Core
 - Wifi (802.11b/g/n up to 150Mbps 2.4GHz)
 - Bluetooth (v4.2 BR/EDR & BLE)
 - 4MB of flash
 - 520KB RAM
- There's lots of competition in this space



Flashing MicroPython:

With the VM:

- Select the VM, plug in SensorNode
 - Should cause it to attach to the VM, if it's not VM -> Removable Devices and attach it
- Helper script (specific to this tutorial)

flash_sensornode.sh

- Sets Serial port (usually /dev/ttyUSB0)
- Fully erases the flash on the ESP32
 - esptool.py --chip esp32 --port
 "\${USBPORT}" erase_flash
- Flashes MicroPython
 - esptool.py --chip esp32 \
 --port "\${USBPORT}" --baud 460800 \
 write_flash -z 0x1000 "\${flash_file}"

Without the VM:

- Serial Drivers
 - Linux: Driver in Most Distros
 - Windows / Mac: Install Silicon Mechanics CP2104 <u>https://www.silabs.com/products/development-to</u> ols/software/usb-to-uart-bridge-vcp-drivers
- Download / Install esptool
 - This requires Python
 - Linux:

distro packages are available

• Windows / Mac:

use pypi to install

- Download MicroPython & Upload it to the board
 - O <u>http://micropython.org/download#esp32</u>
 - esptool.py --chip esp32 \
 --port /dev/ttyUSB0 erase_flash && \
 esptool.py --chip esp32 --port \
 /dev/ttyUSB0 write_flash -z 0x1000 \
 <path to micropython .bin>

Make Sure the SensorNode is 'on'

Blinking Charge Indicator





Helpful tip:

If there's a flashing light on the board it's on (it's the charging indicator light). If it's solid, it's off.

The switch is on the side with the USB port:

- Down = On
- Up = Off

Breaking down the flash commands

esptool.py \ --chip esp32 \ --port /dev/ttyUSB0 \ erase_flash \ 8& \ esptool.py \ --chip esp32 \ --port /dev/ttyUSB0 \ write_flash \ -z 0x1000 \ <path to micropython .bin>

Identifies which chip variant we are dealing with# Identifies which port the serial device is on# Erases the flash area of the chip

(not including the boot loader area)

Identifies which chip variant we are dealing with# Identifies which port the serial device is on# Indicates to write to the flash chip

- # Indicates WHERE on the flash chip to write to
- # What to flash to the chip

What this should look like:

[root@tutorial-base ~]# dmesg | tail -n 8

[...]

[86344.904683] cp210x 2-2.1:1.0: cp210x converter detected [86344.915286] usb 2-2.1: cp210x converter now attached to ttyUSB0

[root@tutorial-base ~]# ./flash_sensornode.sh Flash File: esp32-20190214-v1.10-98-g4daee3170.bin esptool.py v2.7-dev

Serial port /dev/ttyUSB0

Connecting.....

Chip is ESP32D0WDQ6 (revision 1)

Features: WiFi, BT, Dual Core, Coding Scheme None

MAC: 30:ae:a4:86:c7:64

Uploading stub...

Running stub...

Stub running...

Erasing flash (this may take a while)...

Chip erase completed successfully in 4.4s Hard resetting via RTS pin... esptool.pv v2.7-dev Serial port /dev/ttyUSB0 Connecting..... Chip is ESP32D0WDQ6 (revision 1) Features: WiFi, BT, Dual Core, Coding Scheme None MAC: 30:ae:a4:86:c7:64 Uploading stub... Running stub... Stub running... Changing baud rate to 460800 Changed. Configuring flash size... Auto-detected Flash size: 4MB Compressed 1133232 bytes to 714809... Wrote 1133232 bytes (714809 compressed) at 0x00001000 in 18.6 seconds (effective 488.0 kbit/s)... Hash of data verified

Leaving... Hard resetting via RTS pin... [root@tutorial-base ~]#

Open up the serial console

- Minicom:
 - minicom -D /dev/ttyUSB0 --baudrate 115200 (to exit <ctrl>c-q)
- Screen:
 - screen /dev/ttyUSB0 115200n8 (to exit <ctrl>c-A \)
- Windows: use PuTTY

Reset the board



On the serial console...

ets Jun 8 2016 00:22:57

rst:0x1 (POWERON RESET),boot:0x13 (SPI FAST FLASH BOOT) configsip: 0, SPIWP:0xee clk drv:0x00,q drv:0x00,d drv:0x00,cs0 drv:0x00,hd drv:0x00,wp drv:0x00 mode:DIO, clock div:2 load:0x3fff0018,len:4 load:0x3fff001c.len:5060 load:0x40078000.len:8788 ho 0 tail 12 room 4 load:0x40080400.len:6772 entry 0x40081610 (428) cpu start: Pro cpu up. (428) cpu start: Application information: (428) cpu_start: Compile time: 12:32:34 (430) cpu start: Compile date: Feb 14 2019 (436) cpu_start: ESP-IDF: v3.3-beta1-268-g5c88c5996 (442) cpu_start: Single core mode (447) heap init: Initializing. RAM available for dynamic allocation: (454) heap_init: At 3FFAE6E0 len 00001920 (6 KiB): DRAM (460) heap_init: At 3FFB92B0 len 00026D50 (155 KiB): DRAM I (466) heap_init: At 3FFE0440 len 0001FBC0 (126 KiB): D/IRAM (472) heap init: At 40078000 len 00008000 (32 KiB): IRAM (479) heap init: At 40092834 len 0000D7CC (53 KiB): IRAM I (485) cpu_start: Pro cpu start user code (55) cpu start: Starting scheduler on PRO CPU. OSError: [Errno 2] ENOENT MicroPython v1.10-98-g4daee3170 on 2019-02-14; ESP32 module with ESP32 Type "help()" for more information. >>>

Quick Hello World!

>>> print("Hello World!") Hello World!

>>>

Now to Blink an LED!

>>> import machine

>>> led_pin = machine.Pin(0, machine.Pin.OUT)

>>> led_pin.on()

>>> led_pin.off()



Note: You'll quickly find the on() turns the LED off, and off() turns the LED on. To "Fix"

>>> led = machine.Signal(led_pin, invert=True)
>>> led.off()
>>> led.on()

Some interesting things to note

- boot.py
 - executed on every start, good for setting up the board (good place for wifi settings for example)
- main.py
 - Run after boot.py, think of it like the autoexec.bat
- It's possible to upload more files to the board
 - Ampy <u>https://github.com/adafruit/ampy</u>
- Tab completion works in the repl prompt
- <ctrl>+e at the repl prompt puts you into "paste" mode

Disconnect From Serial before trying file transfers!

- Minicom:
 - to exit: <*ctrl*>*c*-*q*
- Screen:
 - to exit: <ctrl>c-A \ y
- Putty:
 - Hit the X and close the application

Where to go from here

Setup Wifi in client mode

- ampy --port /dev/ttyUSB0 get boot.py | tee boot.py
 - # This file is executed on every boot (including wake-boot from deepsleep) #import esp
 - #esp.osdebug(None)
 - #import webrepl
 - #webrepl.start()
- Add to boot.py:
 - # This file is executed on every boot (including wake-boot from deepsleep)
 #import esp
 #esp.osdebug(None)
 #import webrepl
 #webrepl.start()
 import network
 sta = network.WLAN(network.STA_IF)
 sta.active(True)
 sta.connect("ALE", "Penguins")
- ampy --port /dev/ttyUSB0 put boot.py

Re-connect to Serial and check:

- >>> sta.ifconfig()
 ('192.168.123.456', '255.255.255.0', '192.168.123.1', '192.168.123.1')
 >>> sta.status()
 1010
 >>> sta.isconnected()
 True
 >>>

```
- >>> import socket
```

```
>>> addr_info = socket.getaddrinfo("towel.blinkenlights.nl", 23)
>>> addr = addr_info[0][-1]
>>> s = socket.socket()
>>> s.connect(addr)
>>> while True:
... data = s.recv(500)
... print(str(data, 'utf8'), end=")
...
<ctrl>+c will stop the while loop
```

One more thing to note, but not try here...

- Access Point Mode (can be used with client mode at the same time, albeit slowly)
 - >>> ap = network.WLAN(network.AP_IF)
 >>> ap.active(True)
 >>> #ap.config(essid="network-name", authmode=network.AUTH_WPA_WPA2_PSK, password="abcdabcdabcd")
 - Can be added to boot.py, same as the client information

Links to more resources

- <u>https://github.com/unreproducible/tinysnakes</u>
- <u>https://docs.micropython.org/en/latest/esp8266/tutorial/intro.html</u> (note: most of the ideas are the same, the boards ARE different)
- <u>https://boneskull.com/micropython-on-esp32-part-1/</u>
- https://www.cnx-software.com/2017/10/16/esp32-micropython-tutorials/

• Any questions before you start this on your own?

John 'Warthog9' Hawley | <u>warthog9@eaglescrag.net</u> | @warty9