Build the ESP32 Modular Project

Contents

Assembling the Kit	1
Installing Gigabits ESP32 software	11
Set up your Dashboard	32

Introduction

The Gigabits IoT Platform is a backend that makes it easy to acquire, display, store and analyze data from projects. Gigabits IoT Platform's ability to combine a user's data with remote datasets makes it stand out.

This project shows how to use the Gigabits IoT Platform with a several modular sensors from National Control Devices (NCD) produce a simple plant monitor or weather station. It illustrates the use of 8 physical sensors and one button.

Assembling the Kit

Open a web browser to our web page at <u>gigabits.io</u>. Scroll down to the bottom of the page and click "Click here to get your Enterprise Sandbox Kit from NCD"



🕒 Gigabit	ts IoT Simpli	lified!	× C kitcod	e	×	Cart - store	e.ncd.io	×	+					-	
\rightarrow (C 🔒 s	store.ncd.io/c	art/?kc=8fsh									4	ト		D
Apps	😝 Gmail	M Inbox - da	an@gigab 🔽	Classical Music	Radi 🔇 I	rcarle 3911/gigabi	ities M Remoti	e WiFi Setup.		Securely Co	onnectin	Sweather			
	NIC		Shop All	Products ~	IoT Platfo	orms ~ Co	ommunity	Learn		Login /	Register	± ~			
			News	Video						F	۹				
×		ES	SP32 IoT WiFi B	LE Module with	Integrated US	SB			\$17.95		1 +	\$17	95		ł
×	-	120	C Shield for Par	ticle Electron or I	Particle Photo	on with Outward	Facing +5V I2C	Port	\$9.95		1 +	\$9.9	5		
	~	12	C Cable for nod	eLynk Devices a	nd Sensors -	6 inch (152mm)			\$0.95	\$0.00	1	\$0. 9	6	\$0.00	
×		lo ¹	T Training Contr	oller Light Sound	d Sensor Actio	n			\$64.95		1	\$64	95		
	~	12	C Cable for nod	eLynk Devices a	nd Sensors -	6 inch (152mm)			\$0.95	\$0.00	1	\$0. 9	6	\$0.00	
		5 12	Volt 1.25 Amp	Regulated Switc	her Supply				\$24:00	\$19.95	1	\$2 4	00	\$19.95	
×	1	M	Q-2 Propane Bu odule	itane Methane Al	lcohol Gas Se	ensor ADC121C	12-Bit ADC I2C I	Mini	\$27.95		1 +	\$27	95		
	~	12	C Cable for nod	eLynk Devices a	nd Sensors -	3 inch (76mm)			\$0.95	\$0.00	1	\$0. 5	6	\$0.00	
×	1	т	MD26721 Infrare	ed Digital Proximi	ity Detector I2	2C Mini Module			\$25.95		1 +	\$25	95		
	~	12	C Cable for nod	eLynk Devices a	nd Sensors -	3 inch (76mm)			\$0.95	\$0.00	1	\$0. 9	5	\$0.00	
×	1	So	bil Moisture Sen	sor with Analog t	o Digital Con	verter ADC121C	:021		\$25.95		1	\$25	95		
	~	12	C Cable for nod	eLynk Devices a	nd Sensors -	3 inch (76mm)			\$0.95	\$0.00	1	\$0. 9	6	\$0.00	
×	1	тя	SL2561 Light-to-	Digital Converte	r 16-Bit Progr	ammable Gain I	2C Mini Module		\$23.95		1 ‡	\$23	95		
	~	12	C Cable for nod	eLynk Devices a	nd Sensors -	3 inch (76mm)			\$0.95	\$0.00	1	\$0. 9	6	\$0.00	
×	"C) Us	SB Cable Type A	A to Micro 6 Feet	Long				\$3.00		1 +	\$3.0	0		

Modular Figure 1

There are several versions of ESP32 on the market. The one in the kit has been modified by NCD to support the NCD sensors and actuators. That means ESP32's from other sources may not work with this kit.

<u>Kit Step 1:</u> At a minimum, you will need an "ESP32 IoT BLE Module with Integrated USB", a "I2C Shield for Particle Electron or Particle Photon with Outward Facing +5V I2C Port" and one of the remaining boards. We recommend the "IoT Training Controller Light Sound Sensor Action" board with the 12 Volt 1.25 Amp Regulated Switcher Supply". This "Training board" has several types of sensors and actuators. It was designed to introduce new users to IoT. You may choose as many of the remaining boards as you like.

Kit Step 2: Proceed to checkout and order the kit.



<u>Kit Step 3</u>: When the kit arrives, unpack it and make sure that you got everything you ordered. Some of the I2C colorful ribbon cables may be attached to the sensor boards rather than being packaged separately.



Modular Figure 2

<u>Kit Step 4:</u> We need to set an address jumper on the gas sensor, another one on the soil moisture sensor and one on the light sensor. These prevent multiple sensors from attempting to respond to the same address.

<u>Kit Step 4a:</u> Let's start with the soil moisture sensor (the one with the two prongs that you push into dirt). There are six pins labelled "Address:". The pins are arranged in two columns of 3 pins each. The two columns of pins are labeled "0" and "1" as shown in Figure 3. Leave the jumper from the column of pins labelled "1" alone. Move the jumper in column 0 so that it covers the two pins that are furthest from "Address:".





Modular Figure 3



Modular Figure 4

Kit Step 4b: We do something that's almost the same to the Gas sensor.





Modular Figure 5

As we did with the moisture sensor, leave the jumper in the column labelled "1" alone. Connect the jumper in the column labelled "0" so that it covers the two pins closest to the "Address:" text.



Modular Figure 6

Kit Step 4c: The TSL2561 Light sensor also needs a jumper. It has only one column of 3 pins.





Modular Figure 7

Move the jumper so that it covers the two pins closest to "Address".



Modular Figure 8

<u>Kit Step 5:</u> Now that the boards are prepared, let's continue by assembling the ESP32. Remove the piece of black, conductive foam from the ESP32's pins.





Modular Figure 9

<u>Kit Step 6:</u> Press the ESP32 into its adaptor board. The USB micro connector on the ESP32 should be above the I2C connector on the adaptor board. Be firm, but careful not to bend any pins.



Modular Figure 10



Modular Figure 11



<u>Kit Step 7:</u> Use one of the multicolored I2C cables to connect the ESP32 assembly into the training board. Plug the power supply's cable into the Training board. Then plug the power supply into a wall outlet. There are two LEDs next to the Training board's power supply connector. One LED on the training board should light up.



Modular Figure 12

<u>Kit Step 8:</u> Using the multicolored I2C cables, connect the rest of the sensors in a chain. For the sake of consistency, connect the brown wire in each ribbon cable to GND on the board. The order in which boards are chained together should not matter. When this step is complete, two LEDs on the training board and one LED on each of the remaining boards should be lit.





Modular Figure 13



Modular Figure 14



<u>Kit Step 9:</u> Connect the USB A to Micro B cable between the ESP32 Development board and your PC or laptop. The ESP32's USB connector appears to be fairly delicate. Be gentle.



Modular Figure 15



Modular Figure 16

Kit Step 10: The kit has been assembled. Proceed to the next step to install the software.



Installing Gigabits ESP32 software

<u>Software Step 1:</u> Open a web browser and go to "app.gigabits.io".



Software Figure 1

Software Step 2: If you haven't registered with Gigabits yet, use your Beta Key to register now.





Sofware Figure 2

<u>Software Step 3:</u> Once you've registered, go back to app.gigabits.io and sign in.



Gigabits.io 2019 | IoT Simplified

Software Figure 3

<u>Software Step 4:</u> Create a new project by selecting the plus sign. This will create a project named "New Project"



[™] • Gigabits × + ×		- a ×
\leftarrow \rightarrow O \otimes A https://app.gigabits.ia/projects	□ ☆	₫ <i>L</i> & …
Gigabits		Tim Curley -
Projects		
+		
New Project		
Gigabits.io 2019 IoT Simplified		

Software Figure 4

<u>Software Step 5:</u> Select the blue square that's next to your project's name and give your project a nice name by replacing "New Project".



Software Figure 5

<u>Software Step 6:</u> Select "Edit Project" after pressing the project's blue square. This should show you the page used to edit projects.



The second se		- a ×
← → Ů ⋒ A https://app.gigabits.io/projects	□ ☆	¢ & &
©igab <u>its</u>		Tim Curley -
Projects		
+		
Tim's Demo Fim's Demo Edit Project Delete		
Gigabits.io 2019 IoT Simplified		

Software Figure 6

<u>Software Step 7:</u> Every project should have exactly one device (microcontroller). Press the blue "Add Devices" button on the far right. If you can't see that button, make your browser window wider.

💾 🖶 😴 Gigabits	× + ~						-	σ	×
← → O @ A http:	://app.gigabits.io/pro	ojects/d74d134f-7f45-4efe-8	257-f66c81	5659df/edit		□ ☆	☆ ん	É	
Cigabits							Tim	Curley	•
Projects / Tim's Demo / Edit									
Name Tim's Demo Zip Code	Search	Device Key Secret	Submit	Zip Code Gigabits Softv	Devices o 2019 IoT Simplified Vare Figure 7		Add	Device	25

Software Step 8: Add an ESP32 device for your project.





<u>Software Step 9:</u> If you see a mostly blank screen, try clicking on "Gigabits" in the upper left. Then select your project's blue square and select "Edit Project".

<u>Software Step 10:</u> You should now see the page used to edit a project. It should have a device.

🖶 🖅 😨 Gigabits	× + ~						- a ×
	https://app.gigabits.io/projects	/d74d134f-7f45-4efe-8257-f66c815659df/edit				□ ☆	☆ ん ピ …
Cigabits							Tim Curley -
Projects / Tim's Demo / I	Edit						
Name							
Tim's Demo	Search	Submit	D	evices			Add Devices
Zip Code					-		
	Name	Device Key	Secret	Туре	Code		
Save	My New Device	CRu45zA7qSFyz5GDzOEOmmZ	XOjQcQ4CXZiEsQ3KT2KBL9WvRjZ3hpjH	ESP32		dit Delete	
		Gi	gabits.io 2019 IoT Simplified				

Software Figure 9

<u>Software Step 11:</u> Click the "Edit" button in your device's row. This will take you to the page used to edit devices.

🖷 🖅 😨 Gigabits 🛛 🕹 🕂	- ~									- o ×
\leftarrow \rightarrow O \textcircled{a} $\textcircled{https://app.g}$	jigabits.io/pr	ojects/d74d	134f-7f45-4efe-	8257-f66c8	15659df/devices/fd796b	eb-0d9b-401d-be90-1460	ab44cf7f/	'edit	□ ☆	亡 ん ピ …
©igab <u>its</u>										Tim Curley -
Projects / Tim's Demo / Edit / My N	lew Device									
Name										
My New Device	Search			Subm	nit			Sensors		Add Sensors
Device Key										
CRu45zA7qSFyz5GDzOEOm	Name	Index	Data Type	Туре	Minimum Range	Maximum Range				
Secret					-	-				
XOjQcQ4CXZiEsQ3KT2KBL9\										
Generate										
Туре										
ESP32 ~										
Zip Code										
Save										
					Gigabits.io	2019 IoT Simplified				
					Softwa	re Figur	e 1(0		



<u>Software Step 12:</u> Give your device a descriptive name by replacing the default ("My New Device"). Later on, we'll want to display weather data from around the country. Fill in the Zip Code that specifies which weather you'd like to see.

© Gigabits × +									-		×
← → C 🔒 app.gigabits.io/projects	/dc452477-6a5c-	-4b1e-9a75-3	2ba8c619b16/devices	;/089f7a03-5337-4b8e	e-9c7e-9a4a6dfd05fb/edit		☆		人會	D	:
🎹 Apps 🔞 Gmail M Inbox - dan@gigab	Classical M	usic Radi (🗴	rcarle3911/gigabities	M Remote WiFi Setup	💿 Securely Connectin	weather	辩 AWS loT back	end			*
Cigabits									(Guest	•
Projects / Tim's Demo / Edit / Tim's	ESP32										
Name										_	
Tim's ESP32	Search		Submit		Sensors				Add	l Sensoi	s
Device Key				-						-	1
N9pZFW58SArKqH2GidPhKji	Name	Index	Data Type	Туре	Minimum Range	м	aximum Range				
Secret											
J6fbDeuTm8M848ICz3TRhEK											
Generate											
Туре											
ESP32 *											
Zip Code											
94070											
Save Delete											

Software Figure 11

<u>Software Step 13:</u> Add a sensor to your device by clicking on the "Add Sensors" button. This will take you to a page that describes the sensor being added.

🔁 🖅 📀 Gigabits	×	+ ~				-	σ	×
← → ♡ @	A https://	app.gigabits.io/projects/o	174d134f-7f45-4efe-8257-f6	56c815659df/	devices/fd796beb-0d9b-401d-be90-146dab44cf7f/sensors/add	 s≞ l.	, Ŀ	
©igab <u>its</u>						Tin	1 Curle	y -
Projects / Tim's Der	mo / Edit / M	My New Device / Ad	d Sensors					
Search		Submit						
Name	Data Type	Minimum Range	Maximum Range					
General Sensor	NUMERIC			Add				
				C (Gigabits.io 2019 IoT Simplified			

Software Figure 12

Clicking on the "Add" button will take you back to the page used to edit devices. The device will now have one sensor.



 														
 	🗄 😑 📀 Gigabits	× +	+ ~											-
CRu452A7q5Fyz5GDZOEOn CRu452A7q5Fyz5GDZOEOn CRu452A7q5Fyz5GDZOEOn Vojacc4cXZIEsQ3KT2KBL91 Generate Type Esp32 Zip Code gators	÷ → ♡ ⋒	A https://app.	gigabits.io/projects/d7	4d134f-7f45-	4efe-8257-f66c81	15659df/devices/fd796	beb-0d9b-401d-be90-146	idab44cf7f/edit				□ ☆	□ ☆ ☆	□ ☆ ☆ ル
Name Search Submit Sensors Device Key Index Type Type Range Range Idex Device Key CRU45zA7qSFyz5GDzOEOm Index Type Type Range Idex Device Key Idex Idex Type Sensors Idex Idex <td>₿igab<u>its</u></td> <td></td> <td>(</td> <td>Tim C</td>	₿igab <u>its</u>												(Tim C
Name Search Submit Sensors Device Key Image Image Mainimum Maximum Image	Projects / Tim's Demo	/ Edit / Tim's	s ESP32											
Device Key Index Data Type Type Manue Maximum Range Maximum Range Secret My New Sensor 1 NUMERIC General Sensor Index Index Uppe Image Image Image Image Image Image Type Image Image Image Image Image Image Type Image Image Image Image Image Image Type Image Image Image Image Image Image Save Delete Image Image Image Image Image	Name Tim's ESP32		Search		Subm	it		Sensor	5					Add S
CRU452A7q5y25GD2OEOm Name Index Type Type Range Range Secret My New Sensor 1 NUMERIC General Sensor Index Tupe	Device Key													
Secret My New Sensor 1 NUMERIC General Sensor Edit Delete Vpe ESP32 <	CRu45zA7qSFyz5G	DzOEOm	Name	Index	Data Туре	Туре	Range	Range						
XOJQCQ4CXZIESQ3KT2KBL91 MY New Sensor I NUMERIC General Generate Type ESP32 94070 Save Delete	Secret				ALL DATE DIC	Consul			_	_				
Generate Type ESP32 Zip Code 94070 Save Delete	XOjQcQ4CXZiEsQ3	(T2KBL9)	Sensor	1	NUMERIC	Sensor			Edit	Delete				
Type ESP32 Zip Code 94070 Save Delete Gigabits.io 2019 IoT Simplified	Generate													
ESP32 Zip Code 94070 Save Delete Gigabits.io 2019 IoT Simplified	Туре													
Zip Code 94070 Save Delete Gigabits.io 2019 IoT Simplified	ESP32	~												
94070 Save Delete Gigabits.io 2019 IoT Simplified	Zip Code													
Save Delete Gigabits.io 2019 IoT Simplified	94070													
Gigabits.io 2019 IoT Simplified	Save	Delete												
- · · ·						Gigabits.ic	2019 IoT Simplified							

Software Figure 13

<u>Software Step 14:</u> Add another sensor to your device by clicking on the "Add Sensors" button again, then clicking on the "Add" button on the next page again.

🔁 🖅 😨 Gigabits 🛛 🕹 🕂	- ~									- ø ×
\leftarrow \rightarrow O \textcircled{a} https://app.g	gigabits.io/projects/d74	4d134f-7f45-	4efe-8257-f66c81	5659df/devices/fd796l	beb-0d9b-401d-be90-146	5dab44cf7f/edit			□ ☆	作人 ピ …
€ igab <u>its</u>										Tim Curley 🕶
Projects / Tim's Demo / Edit / Tim's	ESP32									
Name										
Tim's ESP32	Search		Submi	t		Sensor	s			Add Sensors
Device Key			-							
CRu45zA7qSFyz5GDzOEOm	Name	Index	Data Type	Туре	Minimum Range	Maximum Range				
Secret							_	_		
XOjQcQ4CXZiEsQ3KT2KBL9\	My New Sensor	1	NUMERIC	General Sensor			Edit	Delete		
Generate	Mr. New	2	NUMERIC	Ganaral			_	_		
Туре	Sensor	2	NOMERIC	Sensor			Edit	Delete		
ESP32										
Zip Code										
94070										
Save										
				Gigabits.io	2019 IoT Simplified	1				
				Softwa	re Figur	e 14				

<u>Software Step 15:</u> Repeat the "Add Sensors, Add" sequence 6 more times for a total of 8 sensors and one actuator.



 			古人日								
Cigabits										Tim Curle	y •
Projects / Tim's Demo / Edit / Tin	n's ESP32										
Name											
Tim's ESP32	Search		Subm	it:		Sens	ors			Add Sense	ors
Device Key											
CRu45zA7qSFyz5GDzOEOm	Name	Index	Data Type	Туре	Minimum Range	Maximum Range					
Secret				-			_	-			
XOjQcQ4CXZiEsQ3KT2KBL9\	My New Sensor	1	NUMERIC	General Sensor			Edit	Delete			
Generate	My New Sensor	2	NUMERIC	General Sensor			Edit	Delete			
ESP32	My New	3	NUMERIC	General			Edit	Delete			
Zip Code	benbor			Jenson .			_				
94070	My New Sensor	4	NUMERIC	General Sensor			Edit	Delete			
Save	My New Sensor	5	NUMERIC	General Sensor			Edit	Delete			
	My New Sensor	6	NUMERIC	General Sensor			Edit	Delete			
	My New Sensor	7	NUMERIC	General Sensor			Edit	Delete			
	My New	8	NUMERIC	General			Edit	Delete			

Software Figure 15

<u>Software Step 16a:</u> Go through all the sensors again. This time, give each sensor a useful name and a color. The name will have two parts. The first part describes the physical sensor type, like "HCPA-5BV-U3" or "TSL2651". The second part is used to identify the sensor, like "HVAC, Room 1". Each sensor type is statically assigned a number called the "sensorIndex" by the system. Each sensor value pushed to the Gigabits server is accompanied by a sensorIndex. That tells the server how to treat the value. Here is the current mapping from sensor type to sensorIndex. This will grow as sensors are added.

There's currently a loophole that we take advantage of. As shown, each sensorIndex corresponds to a particular sensor type. The loophole is that the system doesn't check the sensor type. That means that you can usually replace one sensor with another that has the same class of output (e.g. Numeric) and the system will dutifully display the new sensor's output. You can use the sensor's name to describe the actual sensor output.



Sensor Type	Sensor Index
HCPA-5V-U3, Humidity output	1
HCPA-5C-U3, Temperature output	2
Invert Display button (actuator, not sensor)	3
MPL115A2 Barometric Pressure	4
MQ2 Propane, Butane, Methane, Hydrogen, Alcohol Sensor	5
Soil Moisture Sensor	6
TMD26721 Proximity Sensor	7
TSL2561, Visible Light output	8
TSL2561, Infrared Light output	9

Table 1

<u>Software Step 16b</u>: While you're giving a sensor a name, you should also assign it a nice color. The color will be used to quickly identify gauges and chart lines. The mechanics of assigning a color to a sensor varies between web browsers. In all cases, you click on the color box, then use a browser-specific method to assign a color.

Humidit	ty HVAC Room 1
Index	
1	
Туре	
General	Sensor 👻
Color	
Save	Delete

<u>Software Step 16c:</u> Here's a device with all its sensors. The Invert Display button isn't a sensor. It's a button that causes the little display on the Training board to change from normal video to reverse video and back again. It's our sample actuator.



🗧 😨 Gigabits 🛛 🗙	+ ~									-	٥	×
→ Ů 🏠 🗄 https://app	gigabits.io/projects/d74d134f-7f45-4e	fe-8257-f66	ic815659df/device	s/fd796beb-0d9b-401	d-be90-146dab44cf7f/ed	lit			. ☆	1≡ 1	26	•
Name												
Tim's ESP32	Search	Sut	omit			Sensors				A	ld Sen	sors
Device Key												
CRu45zA7qSFyz5GDzOEOm	Name	Index	Data Type	Туре	Minimum Range	Maximum Range						
Secret XOjQcQ4CXZiEsQ3KT2KBL9\	Humidity HVAC Room 1	1	NUMERIC	General Sensor			Edit	Delete				
Generate	Temperature HVAC Room 2	2	NUMERIC	General Sensor			Edit	Delete				
ESP32	Invert Display Button	3	NUMERIC	General Sensor			Edit	Delete				
94070	Pressure HVAC Room 1	4	NUMERIC	General Sensor			Edit	Delete				
Save	Gas HVAC Room 1	5	NUMERIC	General Sensor			Edit	Delete				
	Soil Sensor Lot 1	6	NUMERIC	General Sensor			Edit	Delete				
	Proximity Gate 1	7	NUMERIC	General Sensor			Edit	Delete				
	Visible Light HVAC Room 1	8	NUMERIC	General Sensor			Edit	Delete				
	Infrared HVAC Room 1	9	NUMERIC	General Sensor			Edit	Delete				

Software Figure 17

<u>Software Step 17:</u> Now that the dashboard is configured, we can start loading software. Start by loading the Arduino IDE. Open a web browser to https://www.arduino.cc/en/main/software.





Software Figure 18

Using the script "Windows Installer, for Windows XP and up", install a copy of the Arduino IDE.

<u>Software Step 18:</u> We need to load the code that links the ESP32 boards with the Arduino IDE. Open File -> Preferences. Put the string <u>https://dl.espressif.com/dl/package_esp32_index.json</u> into the text box labelled "Additional Boards Manager URLs:". Close the Preferences window.



Preferences		×
Settings Network		
Sketchbook location:		
C: \Users \ddavies \Documents	\Arduino	Browse
Editor language:	System Default v (requires restart of Arduino)	
Editor font size:	12	
Interface scale:	Automatic 100 + % (requires restart of Arduino)	
Theme:	Default theme \lor (requires restart of Arduino)	
Show verbose output during:	compilation upload	
Compiler warnings:	None 🗸	
Display line numbers	Enable Code Folding	
Verify code after upload	Use external editor	
Check for updates on sta	rtup Save when verifying or uploading	
Use accessibility features		
Additional Boards Manager UF	RLs (https://dl.espressif.com/dl/package_esp32_index.json	
More preferences can be edit	ed directly in the fire	
C:\Users\ddavies\AppData\Lo	cal\Arduino15\preferences.txt	
(edit only when Arduino is not	running)	
	ОК	Cancel

Software Figure 19

<u>Software Step 19:</u> Select Tools -> Board -> Boards Manager. Search for "esp32 by Espressif Systems". Install this Board package. It will take a few minutes. It may ask if it can modify your PC. Say yes.



🥺 sketch_jan27a /	Arduino 1.8.11	- 0	Boards Manager
File Edit Sketch To	ools Help		Δ
<pre>sketch_jan27a void setup() // put your</pre>	Auto Format Archive Sketch Fix Encoding & Reload Manage Libraries Serial Monitor	Ctrl+T Ctrl+Shift+I Ctrl+Shift+M	Arduino AVR Boards Arduino Yún Arduino Uno Arduino Duemilanove or Diecimila Arduino Nano
} void loop() [Serial Plotter WiFi101 / WiFiNINA Firmware	Ctrl+Shift+L Updater	Arduino Mega or Mega 2560 Arduino Mega ADK Arduino Leonardo
}	Board: "Arduino Leonardo ETH Port Get Board Info Programmer: "AVRISP mkll"	1" 3 	 Arduino Leonardo ETH Arduino Micro Arduino Esplora Arduino Mini Arduino Filiamenti
at jat at jat at jat	Burn Bootloader va.awt.EventDispatchThread va.awt.EventDispatchThread	.pumpEvents(EventD: .run(EventDispatch) Arduino Leon.	Arduino Fio Arduino BT LilyPad Arduino USB LilyPad Arduino Arduino Pro or Pro Mini Arduino NG or older Arduino Robot Control Arduino Robot Motor Arduino Gemma Adafruit Circuit Playground Arduino Yún Mini Arduino Industrial 101
			Linino One Arduino Uno WiFi ESP32 Arduino ESP32 Dev Module ESP32 Wrover Module ESP32 Pico Kit TinyPICO MagicBit Turta IoT Node TTGO LoRa32-OLED V1 TTGO T1

Software Figure 20

.



💿 Boards Manager	×
Type All v esp32	
esp32 by Espressif Systems Boards included in this package: ESP32 Dev Module, WEMOS LoLin32, WEMOS D1 MINI ESP32. More Info	^
1.0.4 ~	Install
	~
	Close

Software Figure 21

Bo	ards Manage		>
ype	All	esp32	
by I Boa ESP	32 Espressif Sy ards included 232 Dev Mod re Info	tems in this package: ile, WEMOS LoLin32, WEMOS D1 MINI ESP32.	^
			Installing
_			
		Downloading tools (3/3). Downloaded 109, 199kb of 1	125,719kb. Cancel

Software Figure 22

<u>Software Step 20:</u> The "esp32" package that you just installed includes many boards. In Tools -> Board, choose the board named "Adafruit ESP32 Feather". In theory, it hardly matters what board you choose. In practice, we haven't found another board that works with the ESP32 device and adaptor from NCD.





Software Figure 23

<u>Software Step 21:</u> Open the Windows Device Manager (On Windows 10, search for "Device Manager". Scroll down to "Ports (COM & LPT). There should be a port with an interesting name, like "Silicon Labs CP210x USB to UART Bridge (COM<a number>)". In this example, the number is 11. Any number other than 1 or 2 should work.



Device Manager	_	Х
<u>File Action View H</u> elp		
> 🥅 Keyboards		^
> 🕼 Mice and other pointing devices		
> 🛄 Monitors		
> 🚽 Network adapters		
> 📃 Portable Devices		
Communications Port (COM1)		
Gommunications Port (COM2)		
🛱 Printer Port (LPT1)		
🛱 Silicon Labs CP210x USB to UART Bridge (COM11)		
> 🚍 Print queues		
> Processors		\sim

Software Figure24

<u>Software Step 22a:</u> That's the Windows COM port that we'll use to communicate with our board. Back in the Arduino IDE, go to Tools -> Port and make sure the port shown is the port from the Device Manager.

File Edit Sketch Tools Help Auto Format Ctrl+T Archive Sketch Serial Coding & Reload Serial I Serial I Serial I SetupPr Board: "Adafruit ESP32 Feather" Serial Plotter Upload Speed: "921600" Serial Nonitor Scheme: "Default" Serial Serial Core Debug Level: "Verbose" Port: "COM11" Serial I Get Board Info Programmer: "AVRISP mkl" Get Board Info Programmer: "AVRISP mkl" SetupGi Jit // publish a message roughly every 5 seconds. 12 // publish a message roughly every 5 seconds. 12 12 setuHCPAData();	00 E	SP32Securel	DemoA	pp Arduino 1.8.11				-	×
Auto Format Ctrl+T Archive Sketch 59 99 901 90 90 101 91 102 92 103 92 104 92 105 92 106 92 107 93 94 108 95 95 96 97 101 98 102 98 99 103 97 90 91 104 92 93 94 94 95 95 96 97 97 98 99 99 90 90 91 91 92 93 94 94 95 95 95 96 96 97 90 91 91 91 92 93 94 94 95 95 95 96 96 96 96 91 91 91 91 91 91 91 92 92	File E	Edit Sketch	Tools	Help		_			
ESP32Secure Archive Sketch 99 Void setu Manage Libraries Ctrl+Shift+I 100 Wire.be Serial Monitor Ctrl+Shift+I 101 Serial. Serial Plotter Ctrl+Shift+I 102 setupDi WiFi101 / WiFiNINA Firmware Updater Serial Plotter Void setup 103 setupDi Board: "Adafruit ESP32 Feather" > 104 setupTS Upload Speed: "921600" > 105 setupTS Upload Speed: "921600" > 106 setupTS Upload Speed: "921600" > 107 WiFi.be Flash Frequency: "80MHz" > 108 setupGi Port: "COM11" Serial ports 109 Serial. Partition Scheme: "Default" > 110 delay(3) Port: "COM11" COM11 111 delay(3) Port: "COM11" COM11 112 SetupGi Programmer: "AVRISP mkli" COM11 114 Programmer: "AVRISP mkli" COM11 COM2 116 // This should be called on very often 1 1		0 🗈 I		Auto Format	Ctrl+T				₽ -
ESP32Secure Fix Encoding & Reload 99 void setu Manage Libraries Ctrl+Shift+1 100 Wire.be Serial Monitor Ctrl+Shift+1 101 Serial. Serial Plotter Ctrl+Shift+L 102 setupDi WiFi101/WiFiNINA Firmware Updater setupME 103 setupME Board: "Adafruit ESP32 Feather" > 104 setupME Board: "Adafruit ESP32 Feather" > 105 setupME Board: "Adafruit ESP32 Feather" > 106 net.set Flash Frequency: "80MHz" > 108 net.set Partition Scheme: "Default" > 109 Mert"COM11" Serial ports COM1 112 Get Board Info COM1 COM1 113 setupGi Get Board Info COM1 114 Programmer: "AVRISP mkli" COM1 COM2 115 J Burn Bootloader COM2 COM1 114 Programmer: "AVRISP mkli" COM2 COM1 COM2 115 J Burn Bootloader COM2 COM1 <td< th=""><th>_</th><th></th><th></th><th>Archive Sketch</th><th></th><th></th><th></th><th></th><th></th></td<>	_			Archive Sketch					
<pre>99 void setu Manage Libraries Ctrl+Shift+1 100 Wire.be Serial Monitor Ctrl+Shift+M 101 Serial. 102 setupD1 103 setupRC WiFi101 / WiFiNINA Firmware Updater 104 setupRF Board: "Adafruit ESP32 Feather" 105 setupFr Board: "Adafruit ESP32 Feather" 106 net.set 107 void loop() { 107 WiPi.she Flash Frequency: "80MHz" 108 Core Debug Level: "Verbose" 110 delay(3 Port: "COM11" 129 Serial. 130 SetupGi Get Board Info 144 15 } 15 Programmer: "AVRISP mkl" 15 } 16 void loop() { 177 void loop() { 177 void loop() { 178 void loop() { 179 void loop() { 179 void loop() { 179 void loop() { 179 void loop() { 170 void loop() { 170 void loop() { 171 void loop() { 172 void loop() { 172 void loop() { 173 setupGi lastMillis > 5000) { 174 lastMillis = millis(); 175 sendECPAData(); 175 void loop() { 175 void loop() { 176 void loop() { 177 void loop()</pre>	ES	P32Secure	F	Fix Encoding & Reload					
<pre>99 Void setu 99 Void setu 90 Void setu 90 Vice.be 90 Vice.be 90 Vice.be 90 Vice.be 90 Vice.be 90 Setup10 90 Setup10 90 Vice.be 90 Setup17 90 Setup17</pre>	98			Manage Libraries	Ctrl+Shift+I				^
<pre>setup11 setup14 setup15 setup15 setup15 setup17 setup16 delay(3 setup16 delay(3 setup16 setup16 setup16 setup17 setup16 setup17 setup16 setup17 setup16 setup1 setup1</pre>	100	Wire he		Serial Monitor	Ctrl+Shift+M				
<pre>setupDi setupAt s</pre>	101	Serial.		Serial Plotter	Ctrl+Shift+1				
103 setupHC WiFi101 / WiFiNINA Firmware Updater 104 setupME Board: "Adafruit ESP32 Feather" 105 setupTS Upload Speed: "921600" 107 WiFi.be Flash Frequency: "80MHz" 108 net.set Flash Frequency: "80MHz" 109 net.set Partition Scheme: "Default" 109 Get Board Info COMI 111 delay(3 Port "COM11" 123 setupG1 Get Board Info 113 setupG1 Programmer: "AVRISP mkI" 114 Programmer: "AVRISP mkI" COM1 115 J Burn Bootloader 117 void loop() { { 114 // This should be called on very often 119 gigabits.tm(); 120 // publish a message roughly every 5 seconds. 121 lastMillis = millis(); sendECPAData(); sendECPAData();	102	setupDi			Curtoninet	-			
<pre>104 setupHE 105 setupHE 106 setupHE 107 setupHE 108 setupFI 109 Serial. Upload Speed: "921600" 100 WiFi.be 108 net.set 109 Serial. Partition Scheme "Default" 100 101 Core Debug Level: "Verhose" 111 122 123 setupGi 124 125 J 125 J 126 Get Board Info 127 void loop() { 128 // This should be called on very often 129 gigabits.run(); 120 121 // publish a message roughly every 5 seconds. 121 iastMillis = millis(); 122 sendECPAData(); </pre>	103	setupHC	<u>۱</u>	WiFi101 / WiFiNINA Firmware Updater					
<pre>105 setupTr Board. "Adafruit ESP32 Feather"</pre>	104	setupMP	-						
<pre>106 setupTS Upload Speed: "921600" 107 WiFi.be Flash Frequency: "80MHz" 109 Serial. Partition Scheme: "Default" 109 Serial. Core Debug Level: "Verbose" 110 delay(3 Port: "COM11" 121 setupGi Get Board Info 122 Get Board Info 123 Port: "COM11" 124 COM11 125 Get Board Info 125 COM1 126 COM1 127 Void loop() { 127 Void loop() { 128 // This should be called on very often 139 gigabits.run(); 129 setuHilis = millis(); 121 setuHilis = millis(); 122 setuHilis = millis(); 123 SetuHilis = millis(); 124 setuHilis = millis(); 125 SetuPS SetUP</pre>	105	setupPr	: E	Board: "Adafruit ESP32 Feather"		>			
107 Wirl.be net.set Flash Frequency: "80MHz" > 109 Serial. Partition Scheme: "Default" > 109 Serial. Port: "COM11" ? 110 delay(3 Port: "COM11" ? 111 delay(3 Port: "COM11" ? 112 SetupGi . COM1 113 setupGi . . 114 Programmer: "AVRISP mkli" . . 115 Burn Bootloader . . 116 V/ This should be called on very often . . 117 void loop() { . . . 118 // publish a message roughly every 5 seconds. . . 120 // publish a message roughly every 5 seconds. . . 121 // publish a message roughly every 5 seconds. . . 122 sendECPAData(); . . .	106	setupTS	i i	Upload Speed: "921600"		>			
<pre>100 met.set 100 serial. Partition Scheme: "Default" 100 delay(3 Port "COM11" 111 delay(3 Port "COM11" 112 Serial ports 113 setupGi 114 Programmer: "AVRISP mkl" 115 } Burn Bootloader 117 void loop() { 116 // This should be called on very often 119 gigabits.run(); 120 // publish a message roughly every 5 seconds. 121 i (millis() - lastMillis > 5000) { 123 lastMillis = millis(); 124 sendECFAData(); </pre>	107	Wiri.de	F	Flash Frequency: "80MHz"		>			
<pre>110 delay(3</pre>	109	Serial.	1	Partition Scheme: "Default"		>			
<pre>111 delay(3 Port "COM11" Serial ports 112 112 setupGi 114 115 ; 116 Programmer: "AVRISP mkll" COM11 115 ; 116 Burn Bootloader 117 void loop() { 118 // This should be called on very often 119 gigabits.run(); 120 121 // publish a message roughly every 5 seconds. 122 if (millis() - lastMillis > 5000) { 123 lastMillis = millis(); 124 sendHCPAData();</pre>	110			Core Debug Level: "Verbose"		>			
<pre>112 113 setupGi 114 115 } 115 } 116 Programmer: "AVRISP mkli" 117 void loop() { 118 // This should be called on very often 119 gigabits.run(); 120 121 // publish a message roughly every 5 seconds. 122 if (millis() - lastMillis > 5000) { 123 lastMillis = millis(); 124 sendHCPAData(); </pre>	111	delay(3	F	Port: "COM11"		-	Serial ports		
<pre>Programmer: "AVRISP mkll" Programmer: "AVRISP mkll" Burn Bootloader COM2 COM11 COM2 COM2 COM1 COM2 COM2 COM2 COM2 COM2 COM2 COM2 COM2</pre>	112	setunGi	(Get Board Info			COM1		
<pre>115 } Programmer "AVRISP mkll" Burn Bootloader 117 void loop() { 118 // This should be called on very often 119 gigabits.run(); 120 121 // publish a message roughly every 5 seconds. 122 if (millis() - lastMillis > 5000) { 123 lastMillis = millis(); 124 sendHCPAData(); 125 </pre>	114	occup-1				~	COM11		
<pre>116 Burn Bootloader 117 void loop() { 117 void loop() { 117 void loop() { 128 // This should be called on very often 119 gigabits.run(); 120 121 // publish a message roughly every 5 seconds. 122 if (millis() - lastMillis > 5000) { 123 lastMillis = millis(); 124 sendHCPAData(); </pre>	115	}	6	Programmer: "AVRISP mkll"		1	COM2		
<pre>117 void loop() { 118 // This should be called on very often 119 gigabits.run(); 120 121 // publish a message roughly every 5 seconds. 122 if (millis() - lastMillis > 5000) { 123 lastMillis = millis(); 124 sendECPAData(); </pre>	116		6	Burn Bootloader		<u> </u>		_	
<pre>118 // This should be called on very often 119 gigabits.run(); 120 121 // publish a message roughly every 5 seconds. 122 if (millis() - lastKillis > 5000) { 123 lastKillis = millis(); 124 sendHCFAData(); </pre>	117	void loop	0 {						
<pre>113 glgarts.tun(); 120 121 // publish a message roughly every 5 seconds. 122 if (millis() - lastMillis > 5000) { 123 lastMillis = millis(); 124 sendHCFAData(); </pre>	118	// This	s shou	ld be called on very often					
<pre>// publish a message roughly every 5 seconds. // publish a message roughly every 5 seconds. // contains a millis > 5000) { lastMillis = millis(); sendHCPAData(); </pre>	120	gigabit	s.run	07					
<pre>122 if (millis() - lastMillis > 5000) { 123 lastMillis = millis(); 124 sendHCPAData(); </pre>	121	// publ	lish a	message roughly every 5 seco	nds.				
<pre>123 lastMillis = millis(); 124 sendHCPAData(); <</pre>	122	if (mil	lis()	- lastMillis > 5000) {					
124 sendHCFAData();	123	lastM	lillis	= millis();					
	124	sendH	ICPADa	ta();					~
		<							>

Software Figure 25

<u>Software Step 22b:</u> If not, press Tools -> Port and select the port from the list of ports. If you don't see your port's name, make sure the ESP32 is turned on and connected. If that's not the problem, try going into the Device Manager again, go down to Ports, right-click on the interesting one and select "Update Driver". If it's still not working, you might installing the drivers from https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers.

<u>Software Step 23a:</u> Use your Arduino IDE to fetch the "Gigabits" library (Sketch -> Include Library -> Library Manager; search for Gigabits). Press its "Install" button.



💿 Library Manager	×
Type All V Topic All V Gigabits	
Gigabits by Robert Carle Version 0.0.3 INSTALLED Gigabits interface for Arduino This library interfaces with the Gigabits API More info Select version v Install	^
	•
	~
Close	:

Software Figure 26

Software Step 23b: Install the "ArduinoJson" library.

💿 Library Manager	\times
Type All v Topic All v ArduinoJson	
ArduinoJson by Benoit Blanchon Version 6.14.1 INSTALLED An efficient and elegant JSON library for Arduino. ArduinoJson supports ✓ serialization, ✓ deserialization, ✓ MessagePack, ✓ fixed allocation, ✓ zero-copy, ✓ streams, and more. It is the most popular Arduino library on GitHub ♥♥♥♥♥. Check out arduinojson.org for a comprehensive documentation. More info Select version ✓	^
 - cloud4rpi-esp-arduino by Cloud4RPi Connect a board to the Cloud4RPi control panel using MQTT - https://cloud4rpi.io. Cloud4RPi client library for ESP8266 and ESP32 based boards. Dependencies: ArduinoJson, PubSubClient. More info 	
Constellation by Sebastien Warin Arduino/ESP library for Constellation 1.8 Arduino/ESP library for Constellation 1.8. This library use the Arduino JSON library (https://github.com/bblanchon/ArduinoJson) (version 5.x) to encode & decode JSON. More info	~
Clos	æ

Software Figure 27

<u>Software Step 23c:</u> Install the "Adafruit_GFX" library.



💿 Library Manager	×
Type All 🗸 Topic All 🗸 Adafruit_GFX	
Adafruit DotStarMatrix by Adafruit Adafruit_GFX-compatible library for DotStar grids Adafruit_GFX-compatible library for DotStar grids More info	^
Adafruit GFX Library by Adafruit Version 1.7.5 INSTALLED Adafruit GFX graphics core library, this is the 'core' class that all our other graphics libraries derive from. Install this library in addition to the display library for your hardware. More info Select version v	
Adafruit ImageReader Library by Adafruit Companion library for Adafruit_GFX and Adafruit_EPD to load images from SD card. Install this library in addition to Adafruit_GFX and the display library for your hardware (e.g. Adafruit_ILI9341), plus the Adafruit_SPIFlash library and SdFat. More info	~
Close	:

Software Figure 28

<u>Software Step 23d:</u> Install the "Adafruit_SSD1306" library. Close the Library Manager window.

💿 Library Manager	Х
Type All V Topic All V Adafruit_SSD 1306	
Adafruit SSD1306 by Adafruit Version 2.1.0 INSTALLED SSD1306 oled driver library for monochrome 128x64 and 128x32 displays SSD1306 oled driver library for monochrome 128x64 and 128x32 displays More info Select version v	^
- Adafruit SSD1306 Wemos Mini OLED by Adafruit + mcauser SSD1306 oled driver library for Wemos D1 Mini OLED shield This is based on the Adafruit library, with additional code added to support the 64x48 display by mcauser. More info	
desklab by Axel Schlindwein, Tobias Schmitt, Jonas Drotleff Implement methods for the use of desklab (www.desk-lab.de) devices. Supports desklab Photometers. You will also have to install Adafruit_SSD1306 and Adafruit-GFX-Library. More info	
Close	•

Software Figure 29

<u>Software Step 24:</u> It's finally time to open the sketch (a program written in Arduino language). If you're writing your own sketch, you can use the rest of these instructions for inspiration. The important parts are including Gigabits.h, calling WiFi.begin with your router's ssid and password, calling gigabits.begin with your device's device key and secret (see below) and calling gigabits.sendRecord each time you want to send a sensor value to the server.



<u>Software Step 25:</u> Using the Arduino IDE, select File -> Examples -> Gigabits. Load the folder named "ESP32SecureDemoApp".



Software Figure 30

At this point, there will be two Arduino IDE windows open. Close the one that we didn't just open.

<u>Software Step 26</u>: We're going to modify the sketch inside that folder. We can't do that from where the folder is now because the directory is read-only. From the Arduino IDE, select "File -> Save As" to write the sketched elsewhere. Saving it on the user's desktop is convenient. That's at C:\Users\<your windows name>\Desktop



Service Servic	32SecureDe t Sketch 1 tw ben ben Recent etchbook amples	emoApp Ardi Fools Help Ctrl+N Ctrl+O	uin	o 1.8.11					-		×	
File Edit Ne Op Op Ski Clo Sav Sav	t Sketch T tw pen pen Recent etchbook amples	Fools Help Ctrl+N Ctrl+O	>									
Ne Op Sko Exa Clo San San	ew oen oen Recent etchbook amples	Ctrl+N Ctrl+O	>									
Op Op Ski Exi Clic San San	oen oen Recent etchbook amples	Ctrl+O	>								101	
Op Sko Exc Clo Sav Sav	een Recent etchbook amples		>									
Ski Exa Clo San San	etchbook amples		12									
Exa Clo San San	amples		,									^
Clo San San			>									
San San	0se	Ctrl+W										
Sav	ve	Ctrl+S										
	ve As	Ctrl+Shift+S										
Pa	ge Setup	Ctrl+Shift+P										
Pri	int	Ctrl+P);								
Pre	eferences	Ctrl+Comma		root_ca); ");							
Qu	uit	Ctrl+Q										
114 115 } 116 117 vc 118 119 120 121 122 123 124 <	// This gigabits // publi if (mill lastMi sendHC) (should be (.run(); sh a messa; is() - las; llis = mil; PAData();	ca: ge tM	<pre>led on ' roughly llis > ' ();</pre>	very ofte every 5 5000) {	n seconds.					>	*
Save Ca	anceled.											
	g at 0x00											^
Writing	g at 0x00											v
<											>	
113			_			_	-	Adafrui	t ESP32 F	eather on	COM11	
-	<i>р</i> тур	e here to se	ear	ch			0	Ħ	9		×	

Software Figure 31

<u>Software Step 27:</u> In the version of ESPSecureDemoApp.ino that you just copied from the user's desktop, scroll down until you get to a line that says "Change these!"



Software Figure 32

Change "ssid" to your router's name. Similarly, replace "pass" with your router's password.



<u>Software Step 28:</u> Way back up above, we declared a project that contained some devices. In the dashboard, go to Projects -> <your project's name> -> Edit. The project's "Device Key" and "Secret" are shown. Copy the Device Key for the project and insert it in place of "devkey" in our sketch. Similarly, replace the sketch's Secret with the project's secret.



Software Figure 33

<u>Software Step 29:</u> Finally, press the Compile and Upload button (in the upper left of the Arduino IDE window, it looks like an arrow facing to the right).



Software Figure 34

This will compile the sketch and start running it. It should start pumping sensor data to the dashboard. You should be able to see this on the Serial Monitor. You can open the Serial Monitor by selecting "Tools -> Serial Monitor"

You've completed the Gigabits setup and are ready to start configuring your dashboard's gauges and charts!



Set up your Dashboard

In this section, we will show how to set up charts, gauges and command buttons.



Dashboard Step 1: Open a web browser to "app.gigabits.io"

Dashboard Figure 1

Sign in. That should take you to the Projects page.



© Gigabits	× M "gbio"	- dan@gigabits.io - Gio 🗙	+	-		×
\leftrightarrow \rightarrow C $$ app.gigab	its.io/projects	07	$\dot{\mathbf{x}}$	۵ 🍾 対		:
👖 Apps 🚳 Gmail M Inbox	- dan@gigab	🔨 Classical Music Radi	s rca	rle3911/gigabi	ities	>>
€ igab <u>its</u>					Guest	•
Projects						
+ Tim's Demo		Dan Test		=•		

Dashboard Figure 2

<u>Dashboard Step 2:</u> Click on the name of the project whose dashboard we'll set up. It's important to click on the project name, not the blue square next to that name. If you're following our example, click on the name "Tim's Demo", not the blue square next to "Tim's Demo".

<u>Dashboard Step 3:</u> On the left side of the User Project page, there are four commands New Chart, New Gauge, New Button and Edit Project. If the commands are chopped off, try making the browser window wider or narrower.





Dashboard Figure 3

<u>Dashboard Step 4:</u> Let's start by making a new gauge. A gauge shows a sensor's current value. Click on the "New Gauge" command. A gauge will appear on the left side of the dashboard. It will be below the lowest gauge, chart or button, which can make it tricky to find because that location is often off the bottom of the screen.





Dashboard Figure 4

The gauge can be resized by dragging the tiny chevron in the bottom/right corner of the gauge. The gauge will have a configuration icon (three horizontal lines). Press the icon to configure the gauge.

<u>Dashboard Step 5:</u> The next page shows the settings for an unconfigured page. One of the first things to notice is that pressing the Delete button deletes the gauge.



New circular - Settings	×
Sensors External	
Select	~
Sensor	
Select	~
Name	
New circular	
Minimum	
Maximum	
Unit of Measurement	
Save De	lete

Dashboard Figure 5

<u>Dashboard Step 6:</u> Here's a configured gauge. It has the on-boarding information discussed above. The Device is from our project. The sensor senses humidity. The name is informative. The Minimum and Maximum bound the range of values shown on the gauge. The Unit of Measurement is used to label the chart axes. The color of the sensor data was chosen when the sensor was set up. Remember to press the Save button when you're done.



Humidit	y, HVAC Room 3198 - Settings	×
Sensors	External	
Device		
User Device	2	~
Sensor		
Humidity S	ensor	~
Name		
Humidity, I	HVAC Room 3198	
Minimum		
0		
Maximum		
100		
Unit of Meas	urement	
RH		
	Save Del	ete

Dashboard Figure 6

<u>Dashboard Step 7:</u> To position the gauge on the dashboard, simply drag it around with the cursor. Dashboard items must be at the top of the dashboard or immediately under another item.





Dashboard Figure 7

<u>Dashboard Step 8:</u> You can also use gauges to display data from external databases. To illustrate this, Gigabits can display weather data. Go to your project and choose New Gauge. Press the new gauge's configuration icon. This time, click on the "External" tab at the top of Settings page. The only choice for "Source" is currently "Open Weather". There are 3 data types, Temperature, Humidity and Barometric Pressure. Pick an informative name. The weather data is for a US Zip Code. That Zip Code was chosen when the project was created. You can also select a nice color for the gauge, minimum and maximum values and a Unit of Measurement. Here is the Settings page for data from Open Weather.



Temperature for Truckee - Settings	×
Sensors External Source	
Open Weather	*
Data Point	
Temperature (Fahrenheit)	~
Name	
Temperature for Truckee	
Color	
Minimum	
-50	
Maximum	
100	
Unit of Measurement	
Deg F	

Dashboard Figure 8

<u>Dashboard Step 9:</u> On to line charts! Press the "New Chart" command. An unconfigured chart will appear on the dashboard. Press its configuration icon. Give the chart a useful name. You can often leave the Unit of Measurement blank since the data being charted normally has different sources.



Temp and Humidity for Greenhouse #5 - Settings							
Chart Settings Name	Sensor Lines	External Lines					
Temp and Humid	ity for Greenhous	e #5					
Unit of Measureme	nt						
Time Format							
h:mm:ss a							
		Save Del	ete				

Dashboard Figure 9

<u>Dashboard Step 10:</u> You can either use a new chart line or modify an existing one. To begin, choose the "Sensor Lines" tab and set "Line" to "Create New Line". Fill in the Device and Sensor items as you did for gauges. Press the Save button. The window does not disappear because it's waiting for another chart line.

Dashboard Step 11: To close the Settings window, click on the "x" in its upper right corner.

<u>Dashboard Step 12</u>: To modify a chart line, open the Settings window, choose the desired line in "Line" and modify the remaining items.

<u>Dashboard Step 13:</u> To delete a chart line, open the chart's Settings, select the chart line in the Line menu and click the Delete button.



Temp and Humidity for Greenhouse #5 ×

	- · · ·	
_	Sottinge	
_	Setunus	

Chart Settings	Sensor Lines	External Lines	
Line			
Temperature Sens	sor		~
Device			
User Device			*
Sensor			
Temperature Sens	sor		~
		Save	Delete

Dashboard Figure 10

Temp and H - Settings	umidity for	Greenhouse #	5 ×
Chart Settings	Sensor Lines	External Lines	
Line			
Create New Line			*
Device			
User Device			*
Sensor			
Humidity Sensor			Ŧ
		Save	Delete
C	Dashboard	Figure 11	

<u>Dashboard Step 14:</u> You can also create chart lines from external data. Press the "External Lines" tab on the Chart Settings page. Fill out "Line" as you did for chart sensors. Fill in Source and Data Point as you for the sensor gauges. Give the line a name and a color.



Temp and Humidity for Greenhouse #5	×
- Settings	

Chart Settings	Sensor Lines	External Lines		
Line				
Temp in Truckee				~
Source				
Open Weather				
Open Weather				
Temperature (Fah	renheit)			Ŧ
Name				
Temp in Truckee				
Color				
		Sa	ive	Delete

Dashboard Figure 12

<u>Dashboard Step 15:</u> And, finally, sending commands. In this demonstration, commands are used to demonstrate that Gigabits can control actuators as well as sensors. This actuator flips the display on the Training board between normal video and reverse video each time the button is pressed.

<u>Dashboard Step 16:</u> Press the New Button item on the dashboard. A new item will appear. Press its configuration icon. Fill in the current project's device. In the current implementation, actuators and sensors have some common code, so we sometimes call both "sensors." That's the case here. Since there is only one actuator (Invert Display command at sensorIndex 3) in the list of "sensors" for this project, the Sensor must be set to Invert Display command.



Invert Display - Settings	×
Device	
User Device	-
Sensor	
Invert Display command	-
Name	
Invert Display	

Save Delete

Dashboard Figure 13

Congratulations! You're ready to explore your data!

