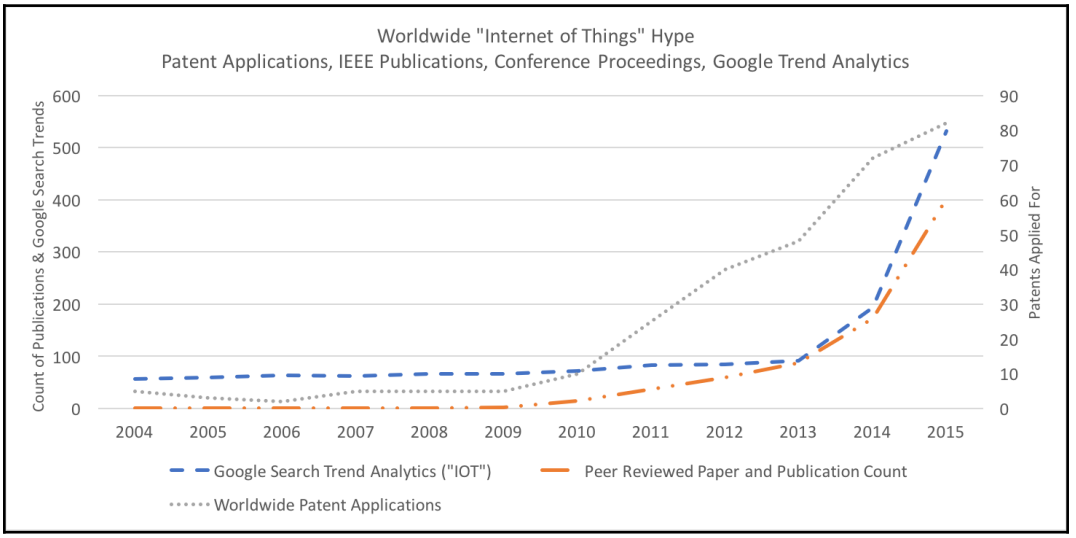
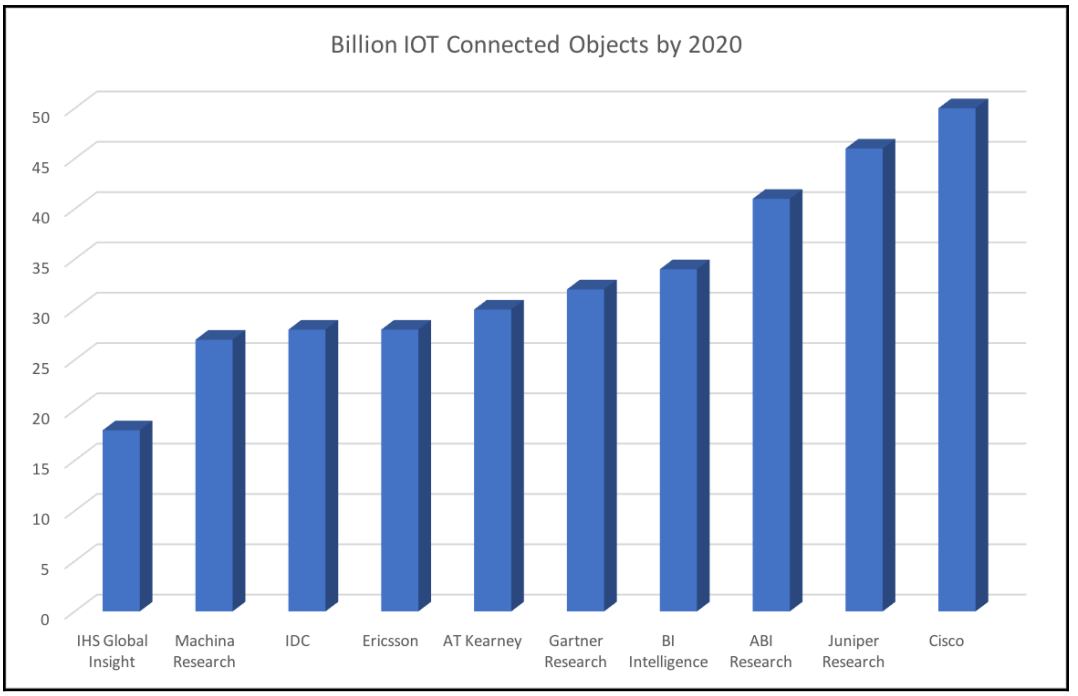
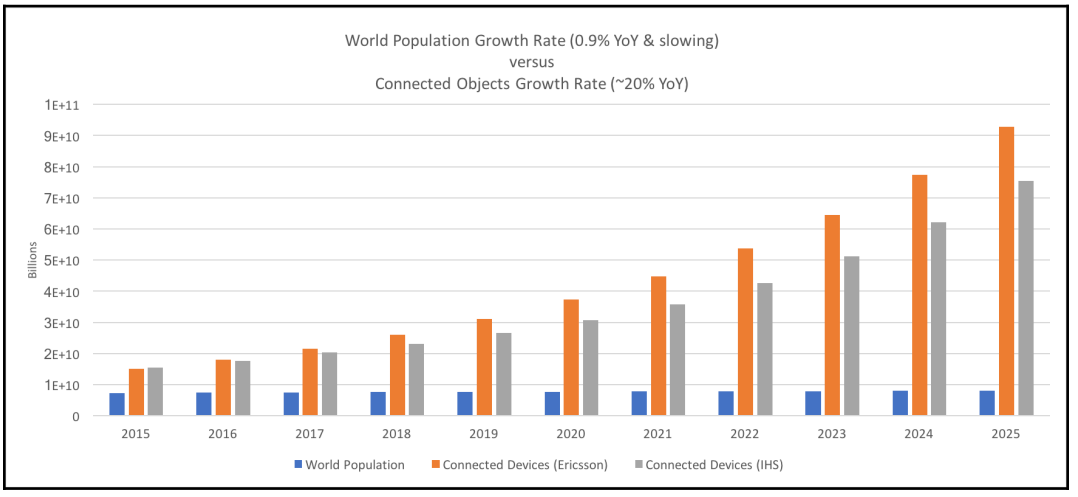
Table of Contents

Graphics

Chapter 1: The IoT Story

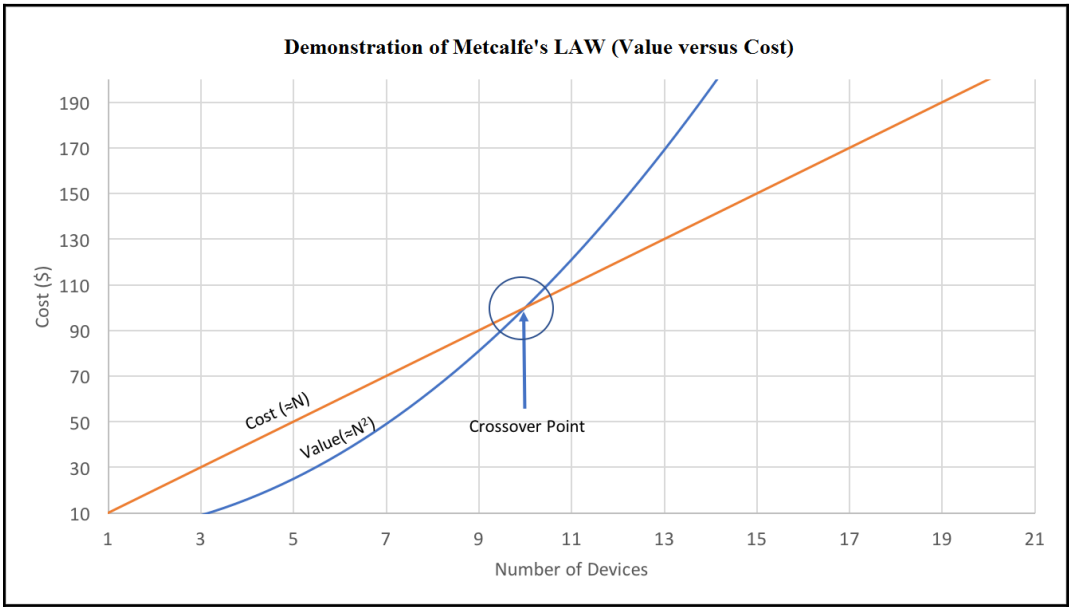



Billion IOT Connected Objects by 2020
50
45
40
35
30
25
20
15
10
5
0
IHS Global Insight
Machina Research
IDC
Ericsson
AT Kearney
Gartner Research
BI Intelligence
ABI Research
Juniper Research
Cisco

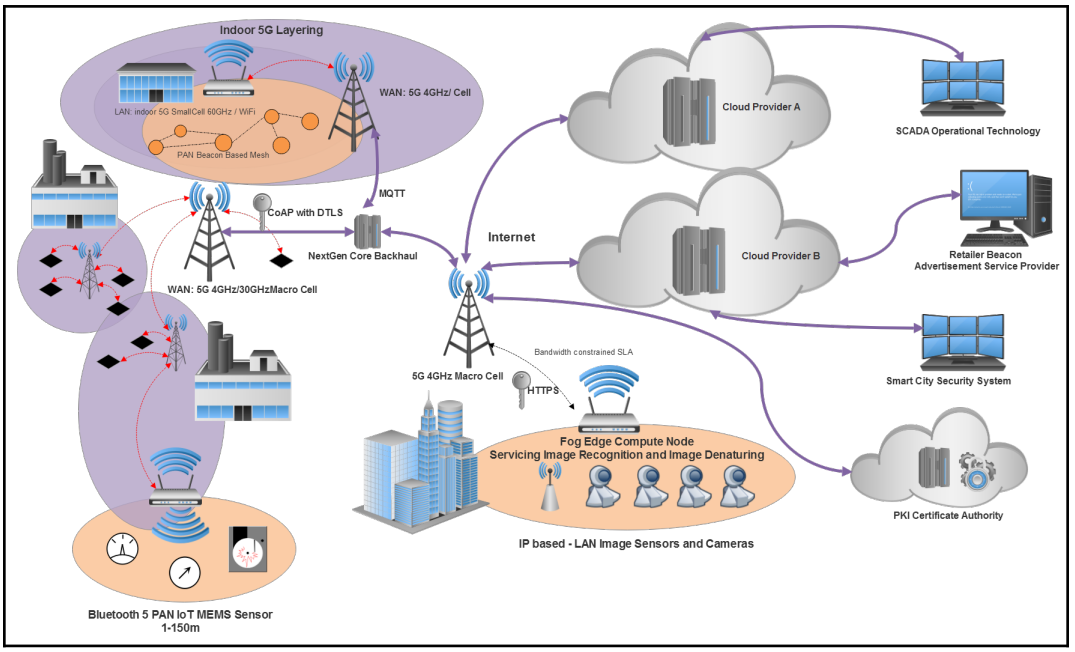

World Population Growth Rate (0.9% YoY & slowing)
versus
Connected Objects Growth Rate (~20% YoY)
Billions
1E+11
9E+10
8E+10
7E+10
6E+10
5E+10
4E+10
3E+10
2E+10
1E+10
0
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
World Population
Connected Devices (Ericsson)
Connected Devices (IHS)

Chapter 2: IoT Architecture and Core IoT Modules

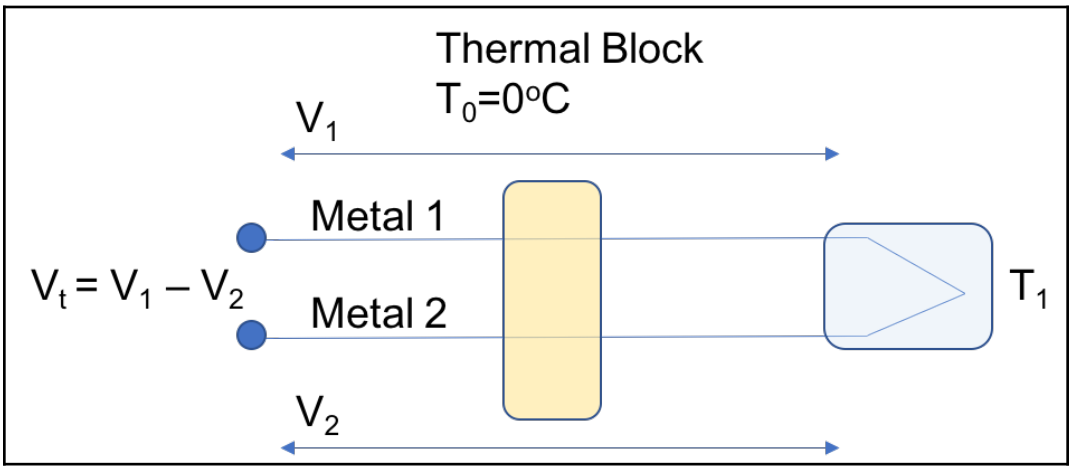
$$V \propto N^2$$

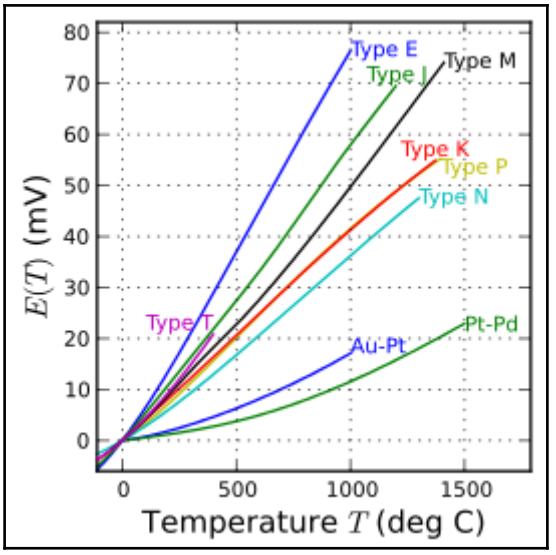


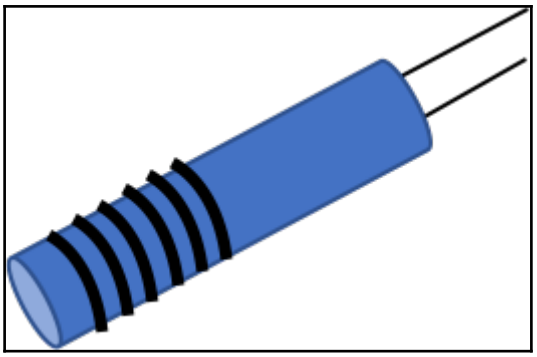
$$\sum_{i=1}^{n} V_{i,j} = \sum_{i=1}^{n} \sum_{k=1}^{m} \frac{B_{i,j,k} - C_{i,j,k}}{(1 + r_k)^{t_k}}$$

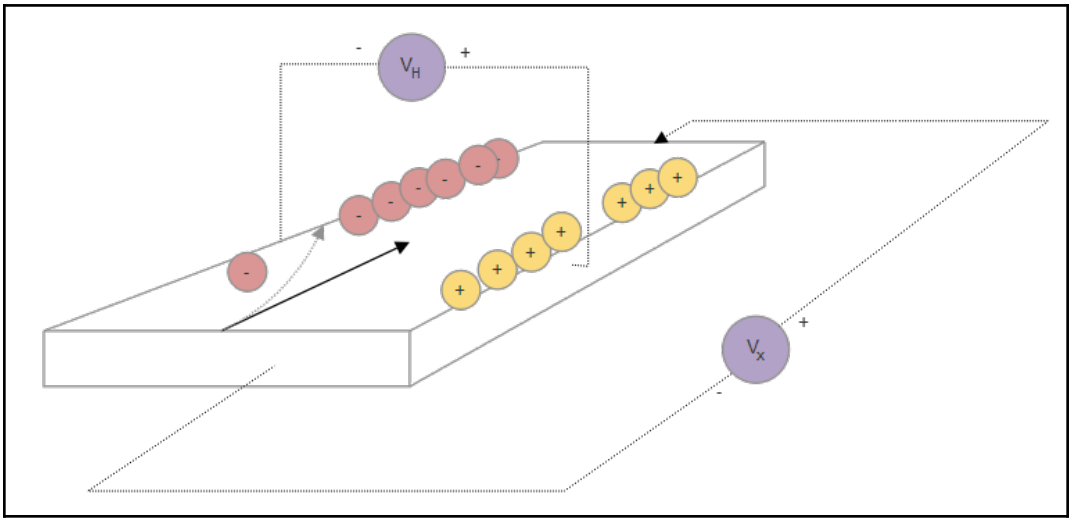

Indoor 5G Layering
LAN: indoor 5G SmallCell 60GHz / WiFi
PAN Beacon Based Mesh
WAN: 5G 4GHz/ Cell
MQTT
CoAP with DTLS
NextGen Core Backhaul
WAN: 5G 4GHz/30GHzMacro Cell
Internet
Cloud Provider A
Cloud Provider B
SCADA Operational Technology
Retailer Beacon
Advertisement Service Provider
Smart City Security System
PKI Certificate Authority
5G 4GHz Macro Cell
Bandwidth constrained SLA
HTTPS
Fog Edge Compute Node
Servicing Image Recognition and Image Denaturing
IP based - LAN Image Sensors and Cameras
Bluetooth 5 PAN IoT MEMS Sensor
1-150m

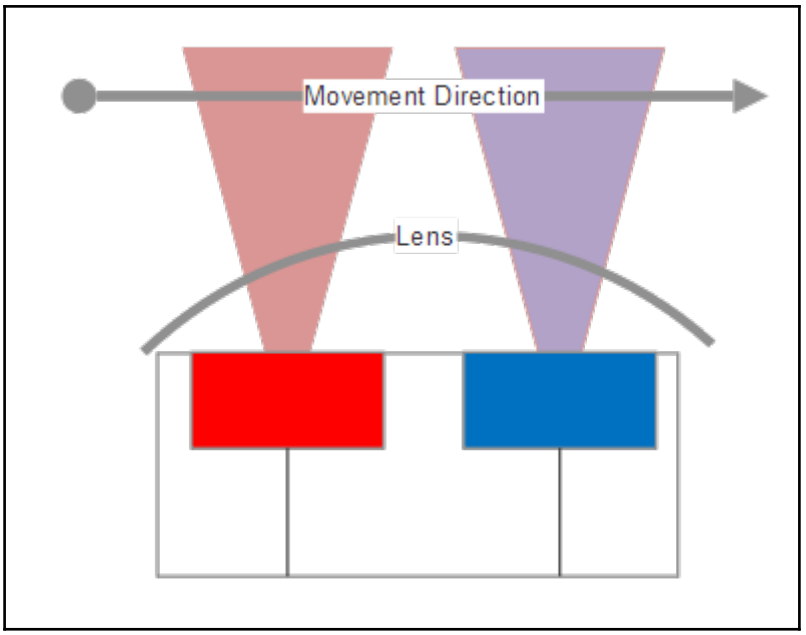
Chapter 3: Sensors, Endpoints, and Power Systems

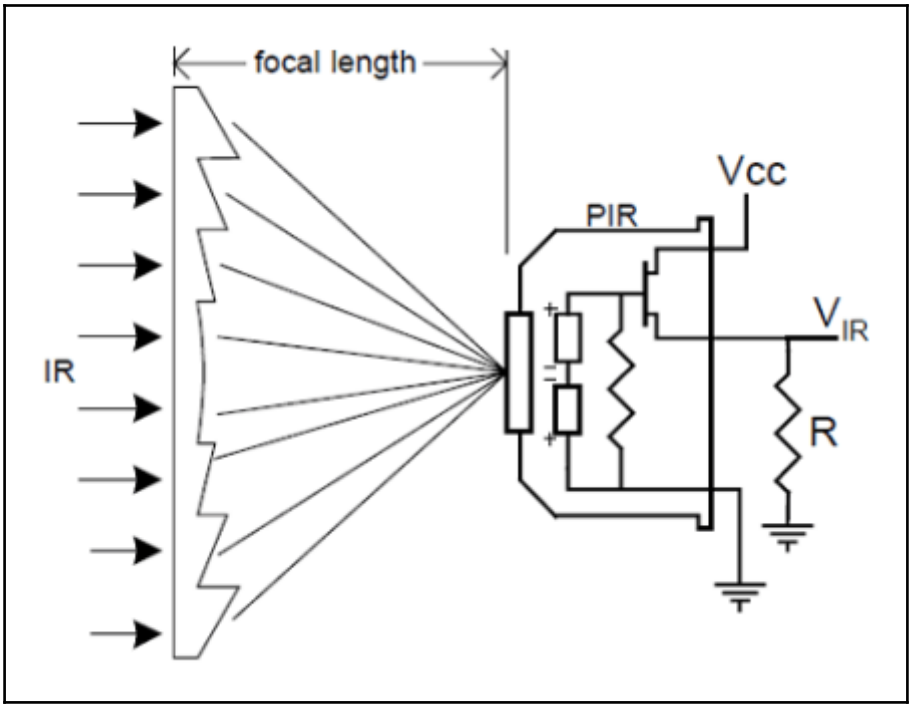




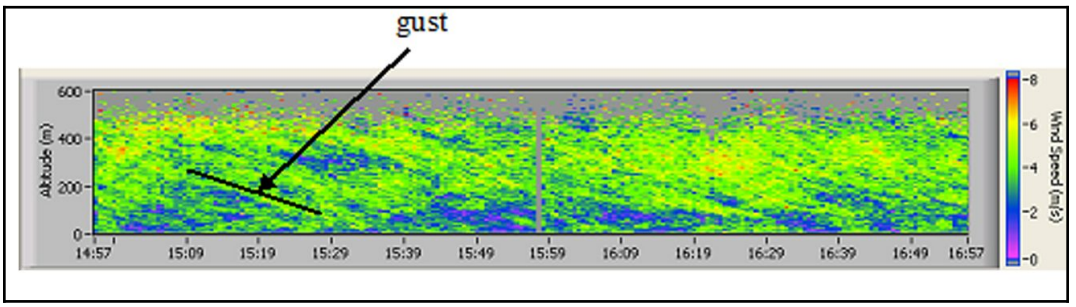


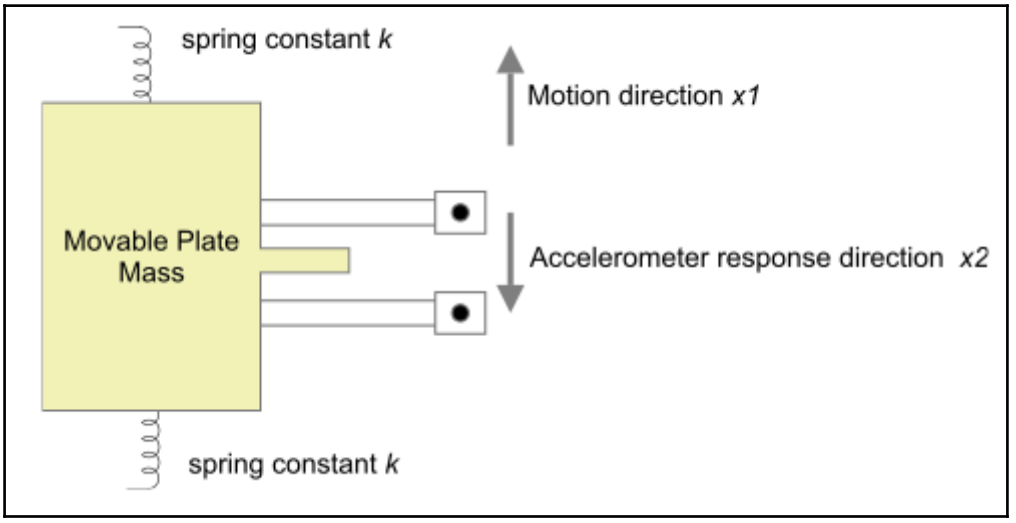

-
V_H
+
-
+
V_x
+
-

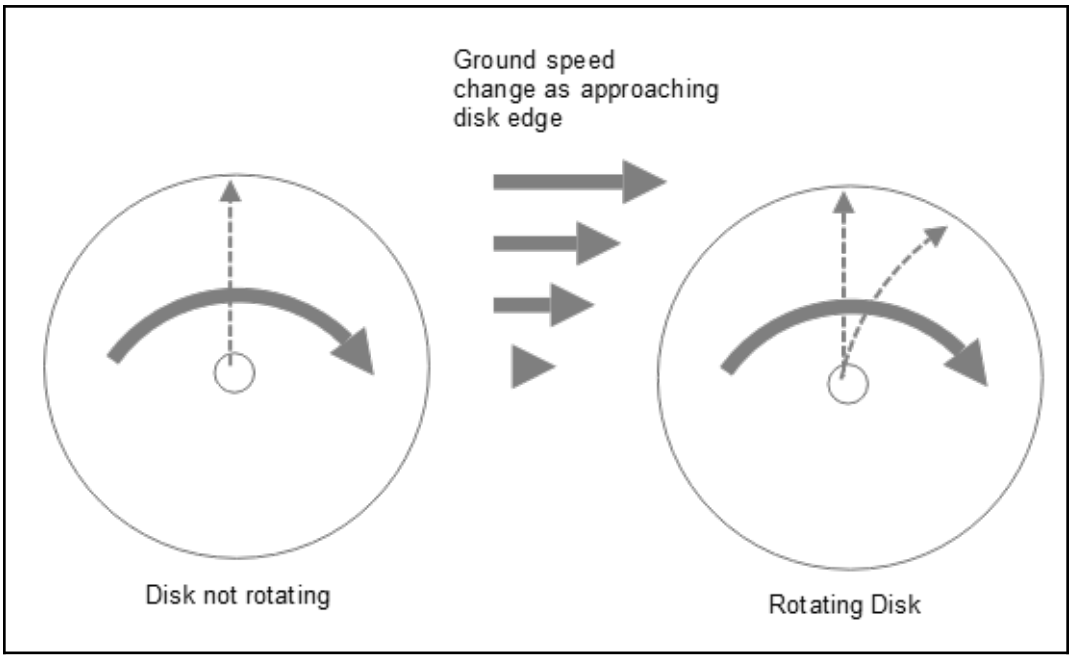

Movement Direction
Lens


focal length
IR
Vcc
PIR
V_{IR}
R

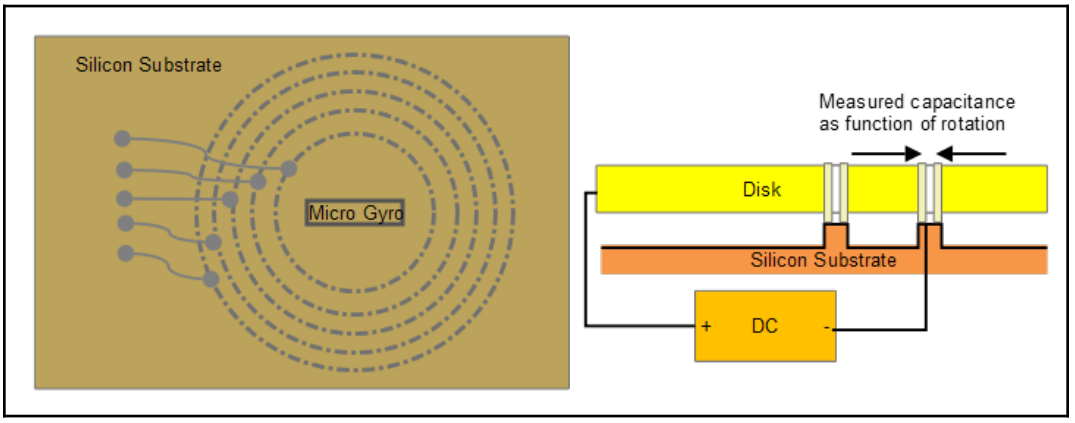
$$Distance = \frac{(Speed\ of\ Light \times Time\ of\ Flight)}{2}$$

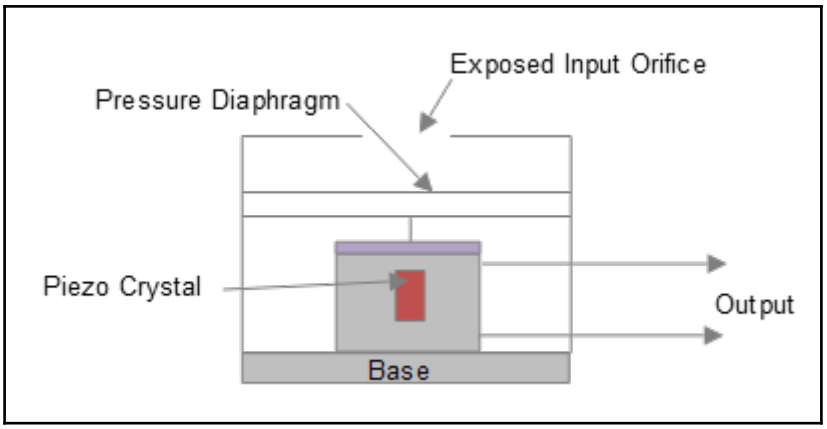


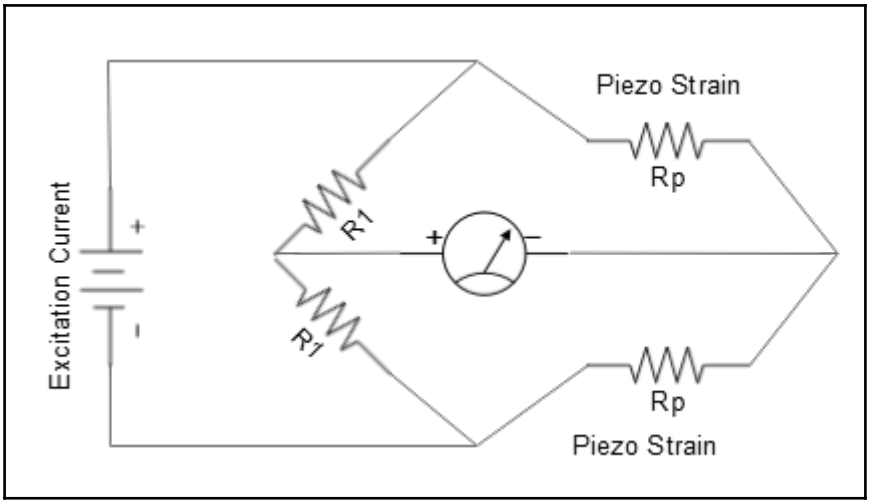


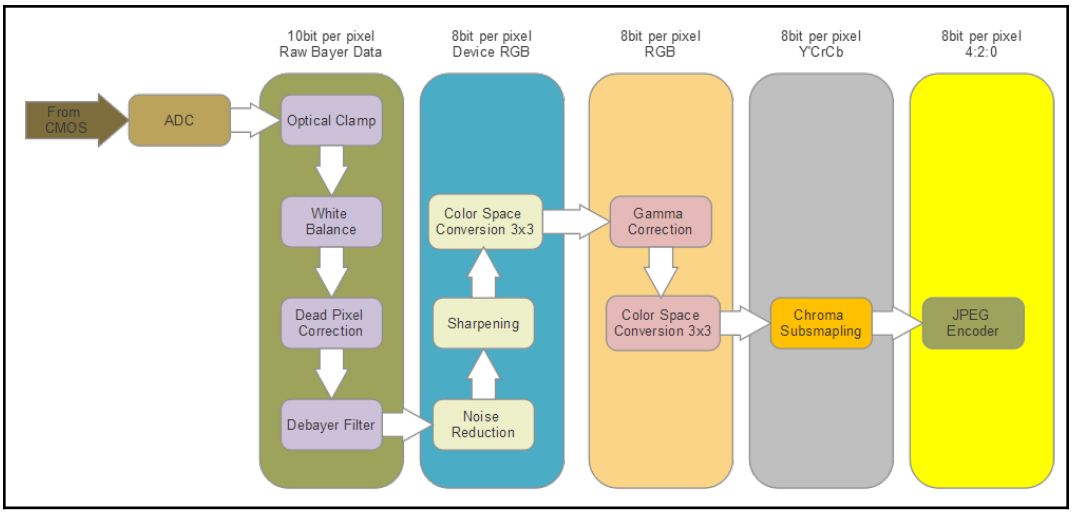


$$a = -2\omega \times v$$

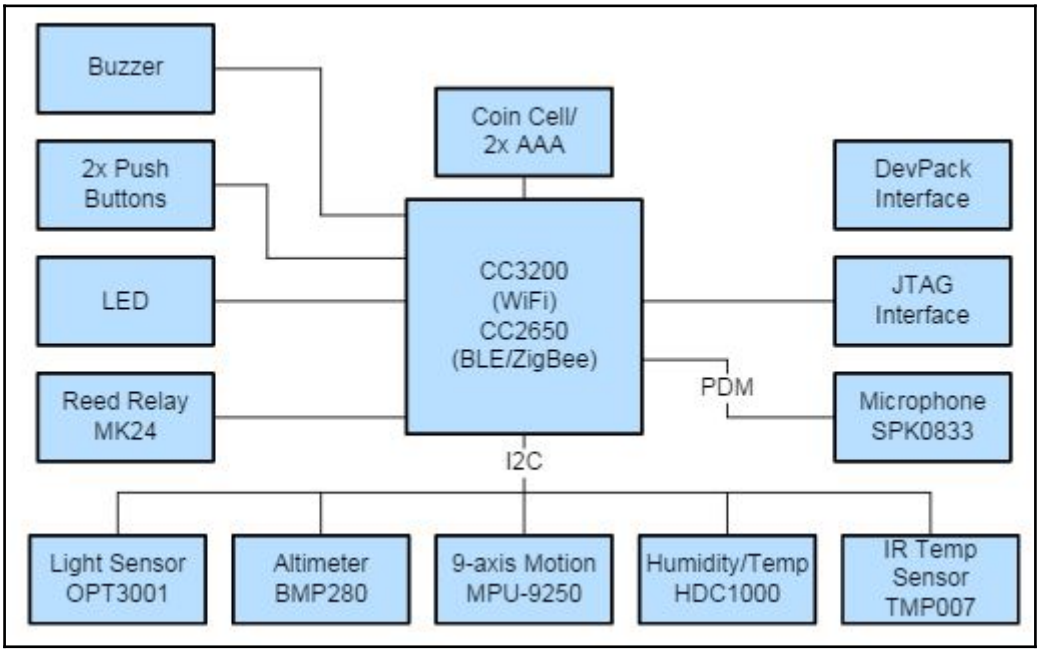


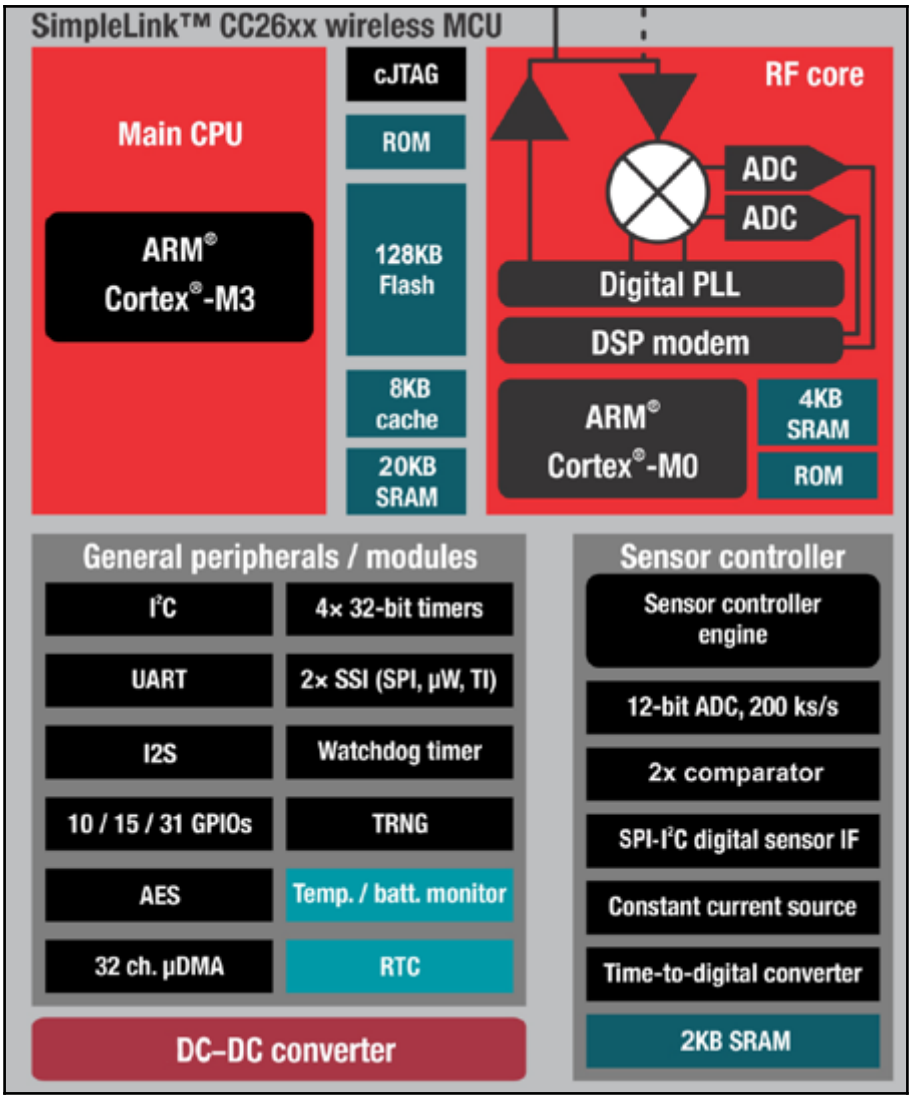

Exposed Input Orifice
Pressure Diaphragm
Piezo Crystal
Output
Base

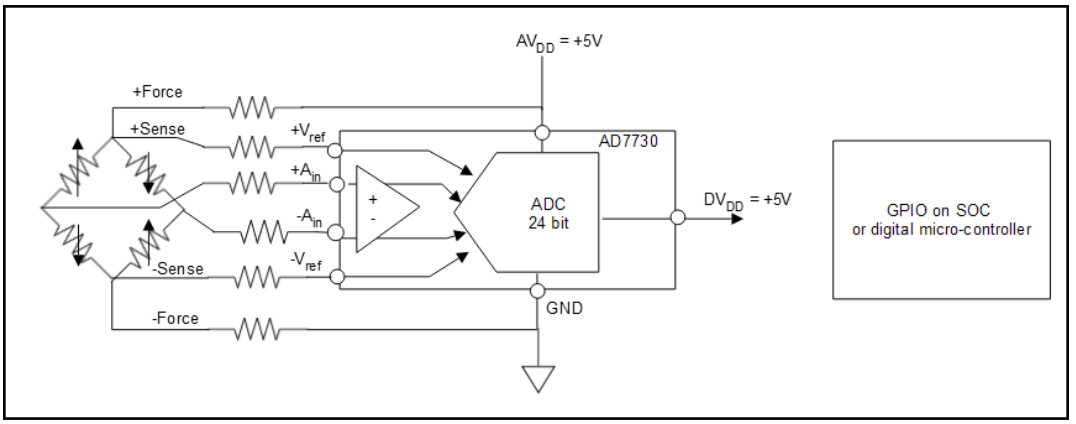

Piezo Strain
Rp
R1
+
−
Excitation Current
+
-
R1
Rp
Piezo Strain

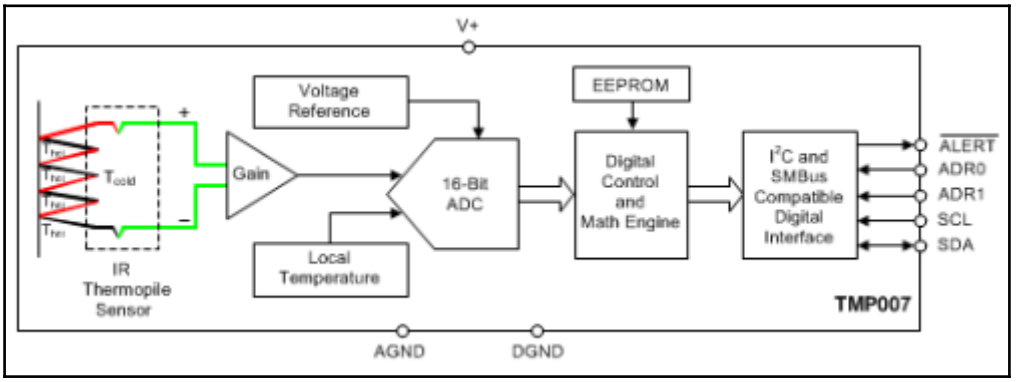


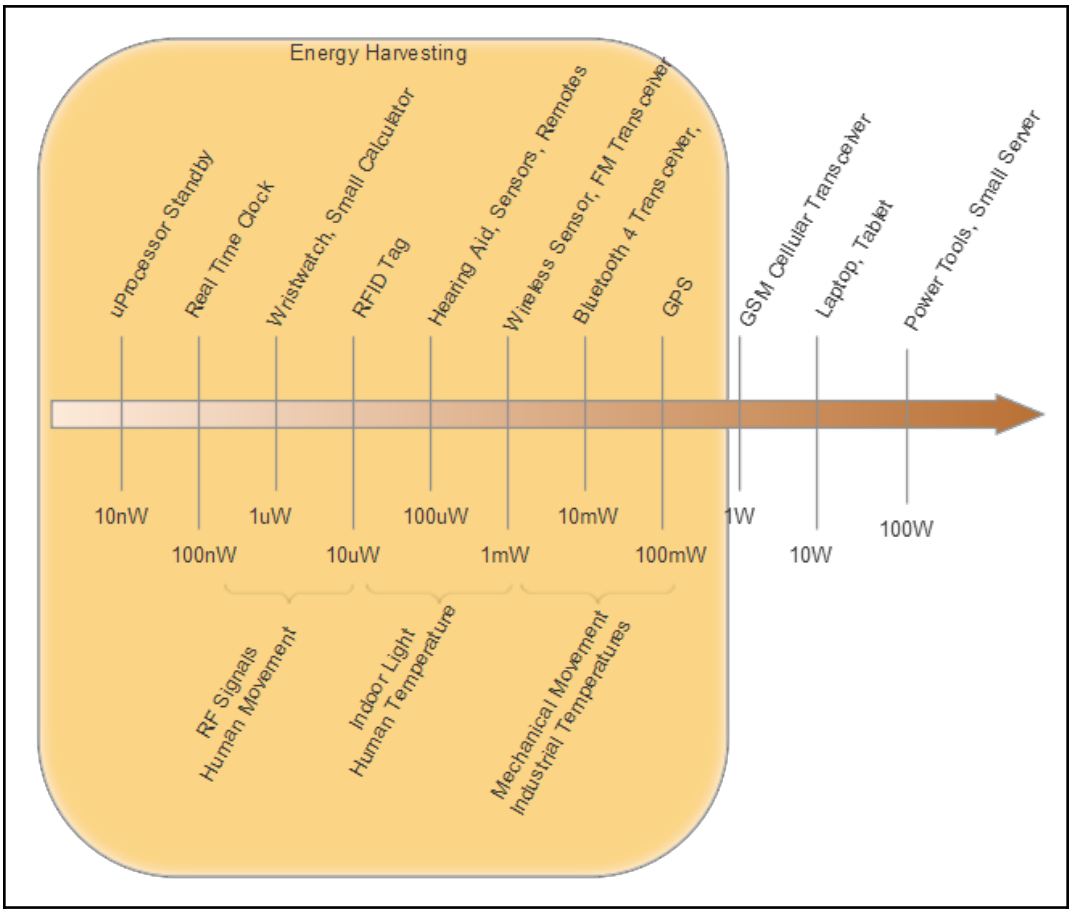
$$x_3 = (\sigma_1^{-2} + \sigma_2^{-2})^{-1}(\sigma_1^{-2}x_1 + \sigma_2^{-2}x_2)$$

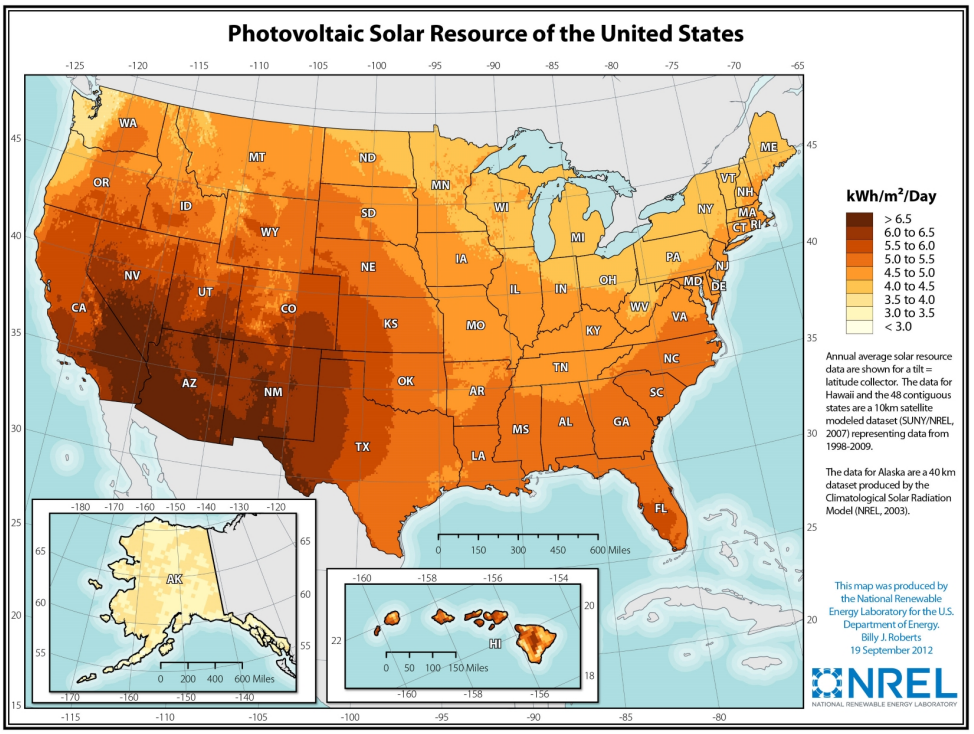



SimpleLink™ CC26xx wireless MCU
Main CPU
ARM® Cortex®-M3
cJTAG
ROM
128KB Flash
8KB cache
20KB SRAM
RF core
ADC
ADC
Digital PLL
DSP modem
ARM® Cortex®-M0
4KB SRAM
ROM
General peripherals / modules
I²C
4× 32-bit timers
UART
2× SSI (SPI, µW, TI)
I2S
Watchdog timer
10 / 15 / 31 GPIOs
TRNG
AES
Temp. / batt. monitor
32 ch. µDMA
RTC
DC–DC converter
Sensor controller
Sensor controller engine
12-bit ADC, 200 ks/s
2x comparator
SPI-I²C digital sensor IF
Constant current source
Time-to-digital converter
2KB SRAM


AVDD = +5V
+Force
+Sense
+Vref
+Ain
-Ain
-Sense
-Vref
-Force
AD7730
ADC
24 bit
DVDD = +5V
GND
GPIO on SOC
or digital micro-controller


V+
Voltage
Reference
EEPROM
Gain
16-Bit
ADC
Digital
Control
and
Math Engine
I2C and
SMBus
Compatible
Digital
Interface
ALERT
ADR0
ADR1
SCL
SDA
Tcold
IR
Thermopile
Sensor
Local
Temperature
TMP007
AGND
DGND


Energy Harvesting
uProcessor Standby
Real Time Clock
Wristwatch, Small Calculator
RFID Tag
Hearing Aid, Sensors, Remotes
Wireless Sensor, FM Transceiver
Bluetooth 4 Transceiver,
GPS
GSM Cellular Transceiver
Laptop, Tablet
Power Tools, Small Server
10nW
100nW
1uW
10uW
100uW
1mW
10mW
100mW
1W
10W
100W
RF Signals
Human Movement
Indoor Light
Human Temperature
Mechanical Movement
Industrial Temperatures

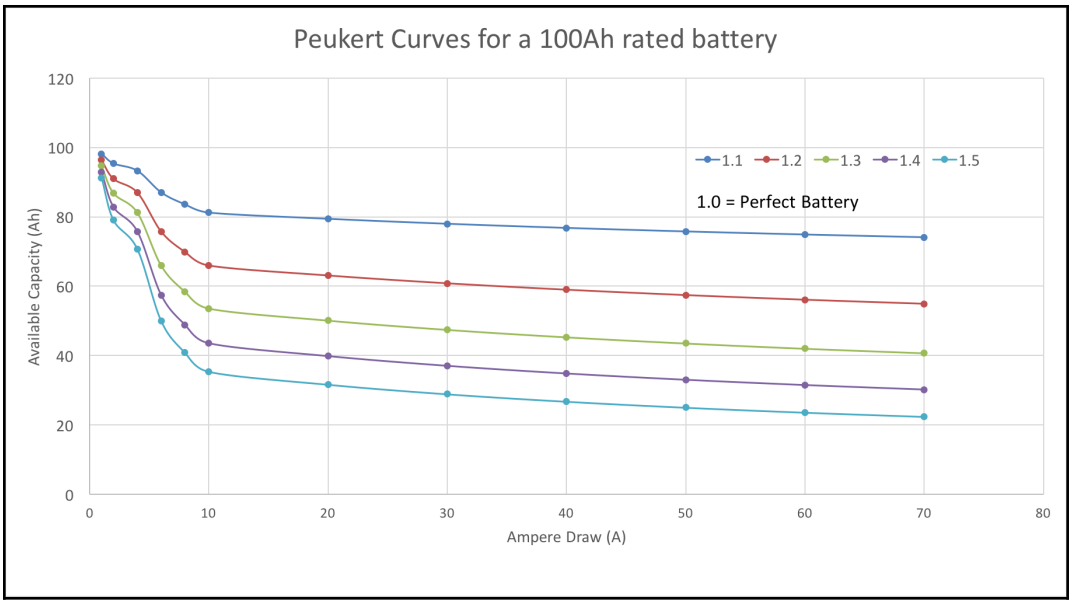


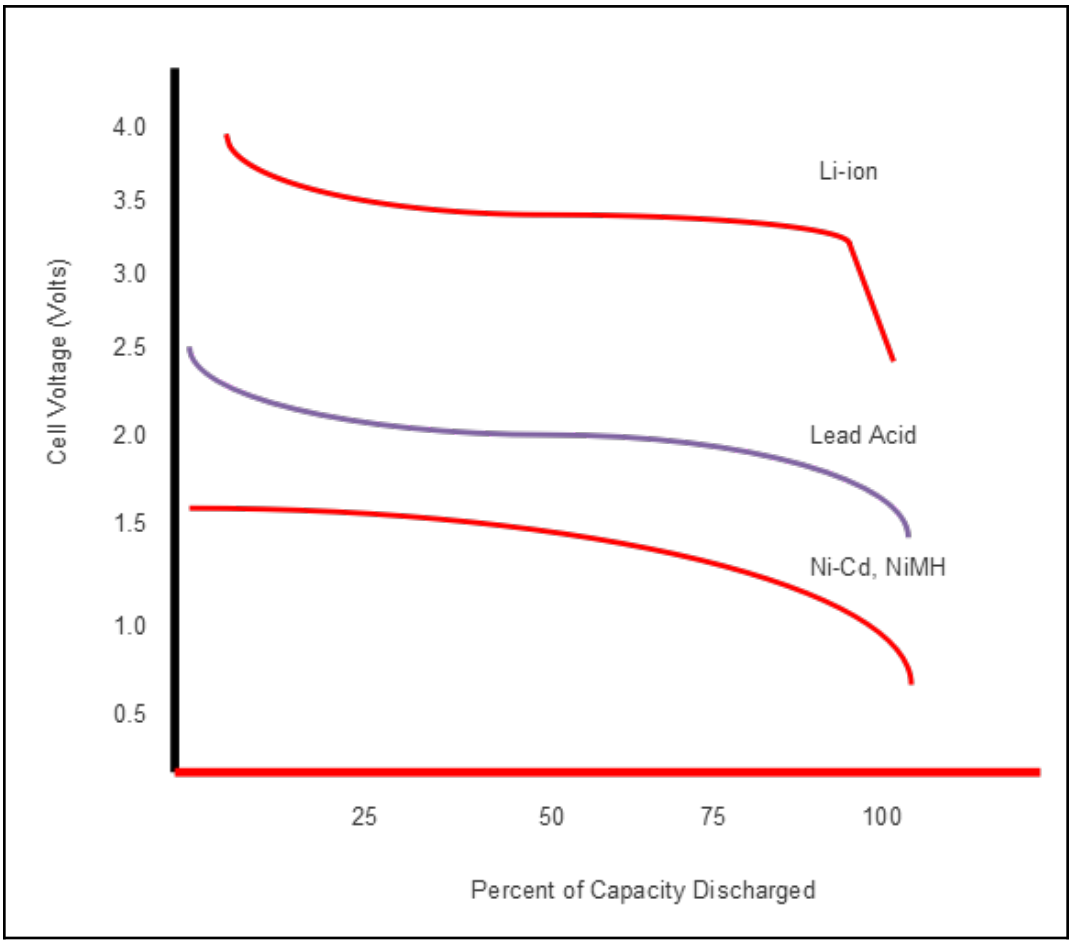
$$E = \frac{1}{2}QV^2 = \frac{Q^2}{2C}$$

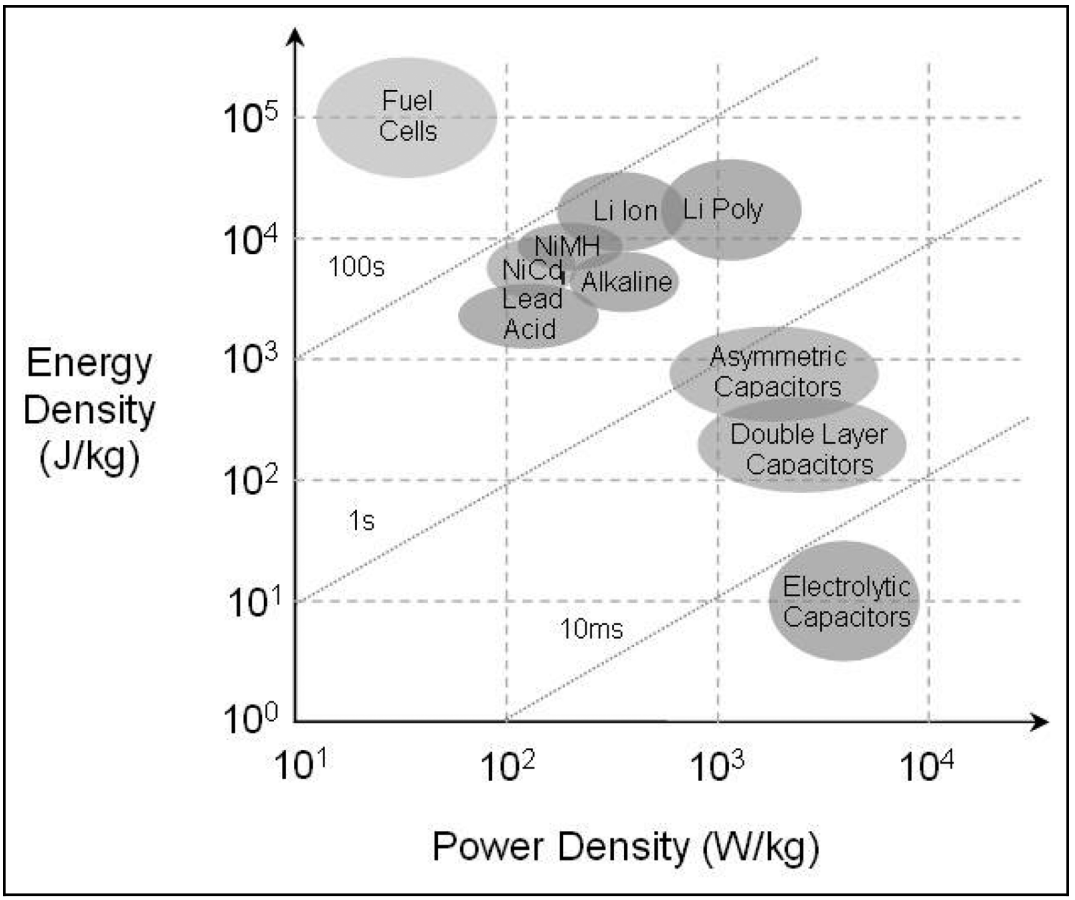
$$C = \varepsilon_0 L_w / d$$

$$V = \int_{T_L}^{T_H} S_1(T) - S_2(T) dT$$

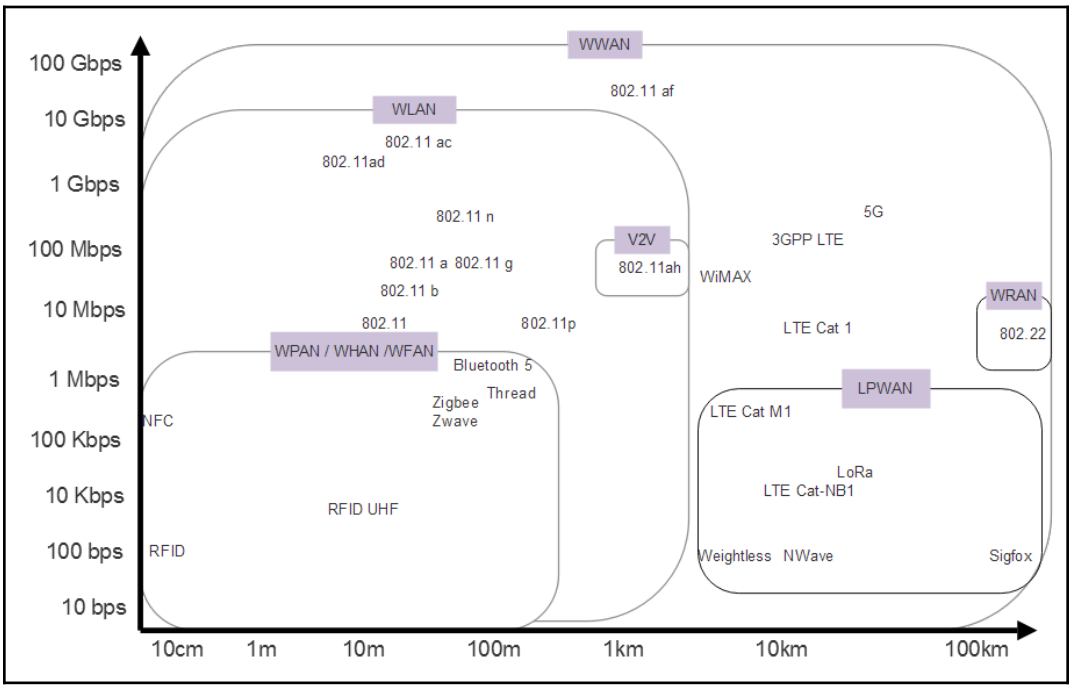
$$t = \frac{C_p}{I^n}$$




4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
Cell Voltage (Volts)
Li-ion
Lead Acid
Ni-Cd, NiMH
25
50
75
100
Percent of Capacity Discharged

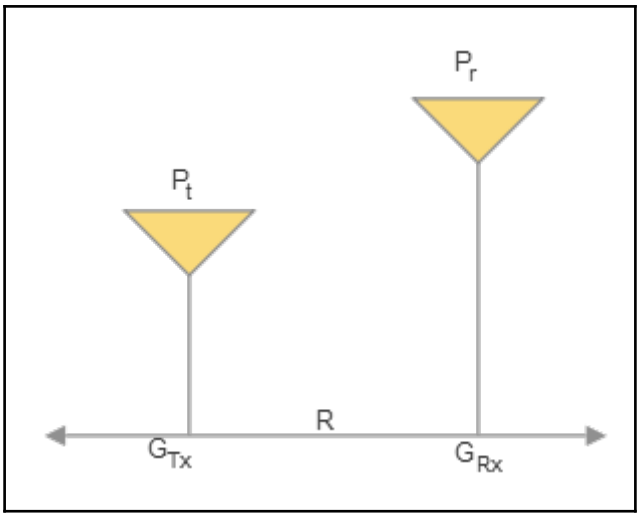

Energy Density (J/kg)
10^5
10^4
10^3
10^2
10^1
10^0
Fuel Cells
Li Ion
Li Poly
NiMH
NiCd
Alkaline
Lead Acid
100s
1s
10ms
Asymmetric Capacitors
Double Layer Capacitors
Electrolytic Capacitors
10^1
10^2
10^3
10^4
Power Density (W/kg)

Chapter 4: Communications and Information Theory



$$P_r = P_t G_{Tx} G_{Rx} \frac{\lambda^2}{(4\pi R)^2}$$

$$P_r = P_t + G_{Tx} + G_{Rx} + 20log_{10}\left(\frac{\lambda}{4\pi R}\right)$$

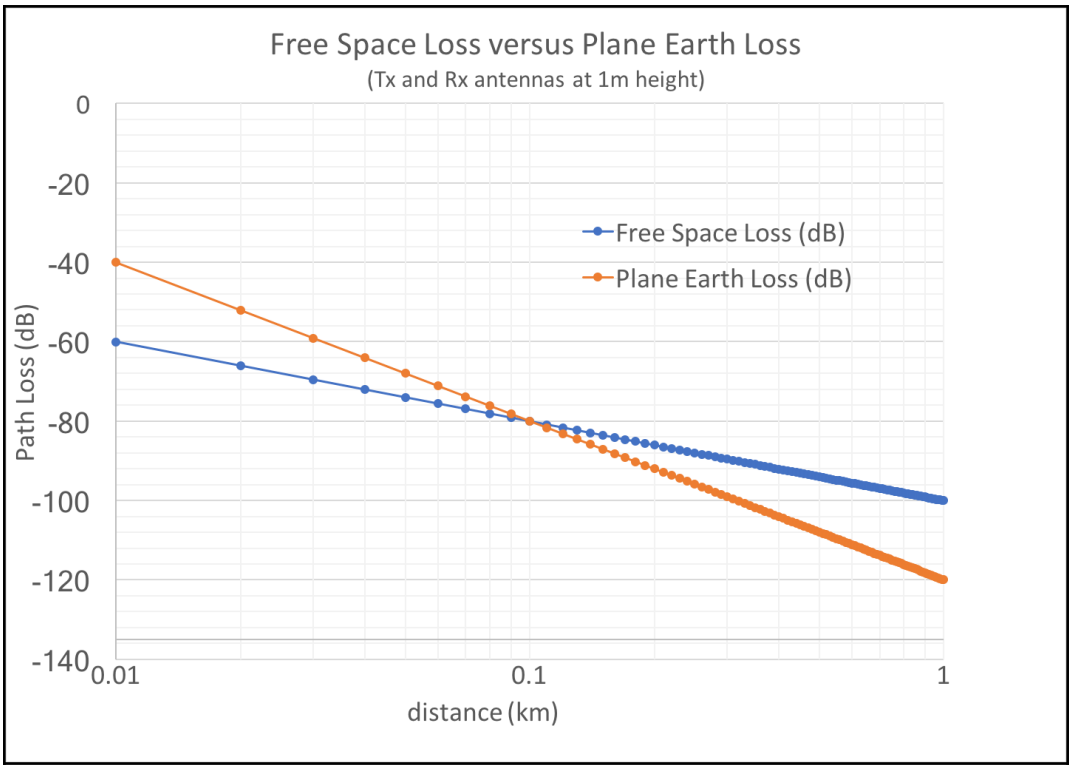


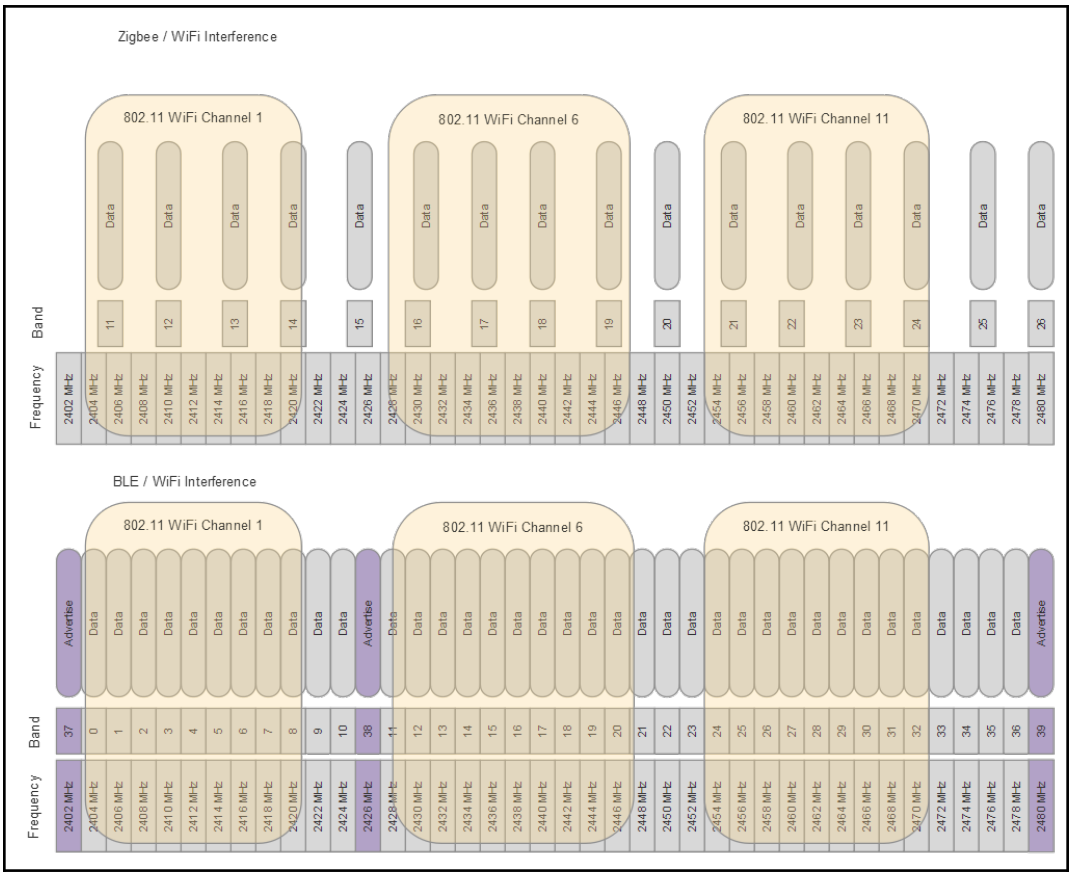
$$LinkBudget = \frac{Tx\ Power}{Rx\ sensitivity\ level}$$

$$Receiver\ Power(dB) = Transmitted\ Power(dB) + Gains(dB) - Losses(dB)$$

$$\begin{aligned} FSPL(dB) &= 10log_{10}\left(\left(\frac{4\pi Rf}{c}\right)^2\right) \\ &= 20log_{10}\left(\frac{4\pi Rf}{c}\right) \\ &= 20log_{10}\left(R\right) + 20log_{10}\left(f\right) + 20log_{10}\left(\frac{4\pi}{c}\right) \\ &= 20log_{10}\left(R\right) + 20log_{10}\left(f\right) - 147.55 \end{aligned}$$

$$\frac{Pr}{Pt} = L_{plane\ earth\ loss} \approx \left(\frac{\lambda}{4\pi R} k \frac{2h_t h_r}{R}\right) \approx \frac{{h_t}^2 {h_r}^2}{R^4}\ where\ k = \frac{2\pi}{\lambda}$$


Free Space Loss versus Plane Earth Loss
(Tx and Rx antennas at 1m height)
0
-20
-40
-60
-80
-100
-120
-140
Path Loss (dB)
0.01
0.1
1
distance (km)
Free Space Loss (dB)
Plane Earth Loss (dB)



$$f_p \leq 2B$$

$$M = 1 + \frac{A}{\triangle V}$$

$$R = f_p log_2(M)$$

$$R \leq 2Blog_2(M)$$

$$C = Blog_2(1 + \frac{S}{N})$$

$$C = B \times n \times log_2(1 + \frac{S}{N})$$

$$C = Blog_2(1 + \frac{S}{N})$$

$$200 = 5000 \times log_2(1 + \frac{S}{N})$$

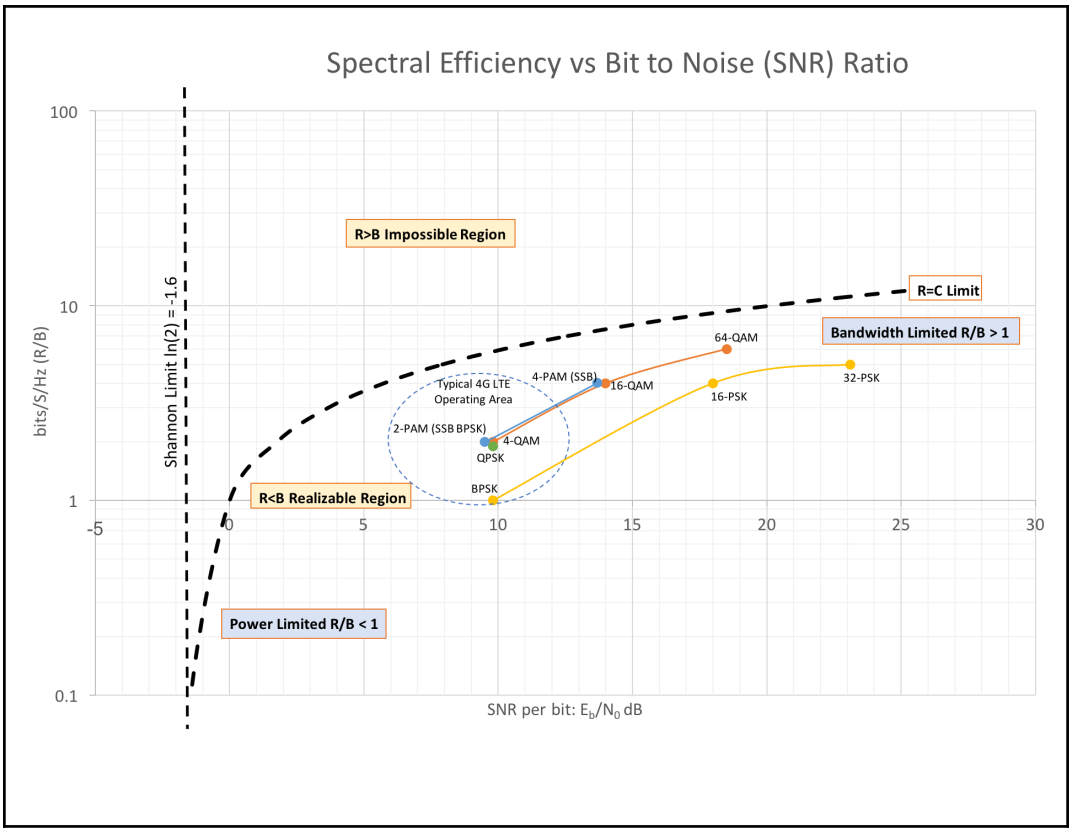
$$\frac{S}{N} = 0.028$$

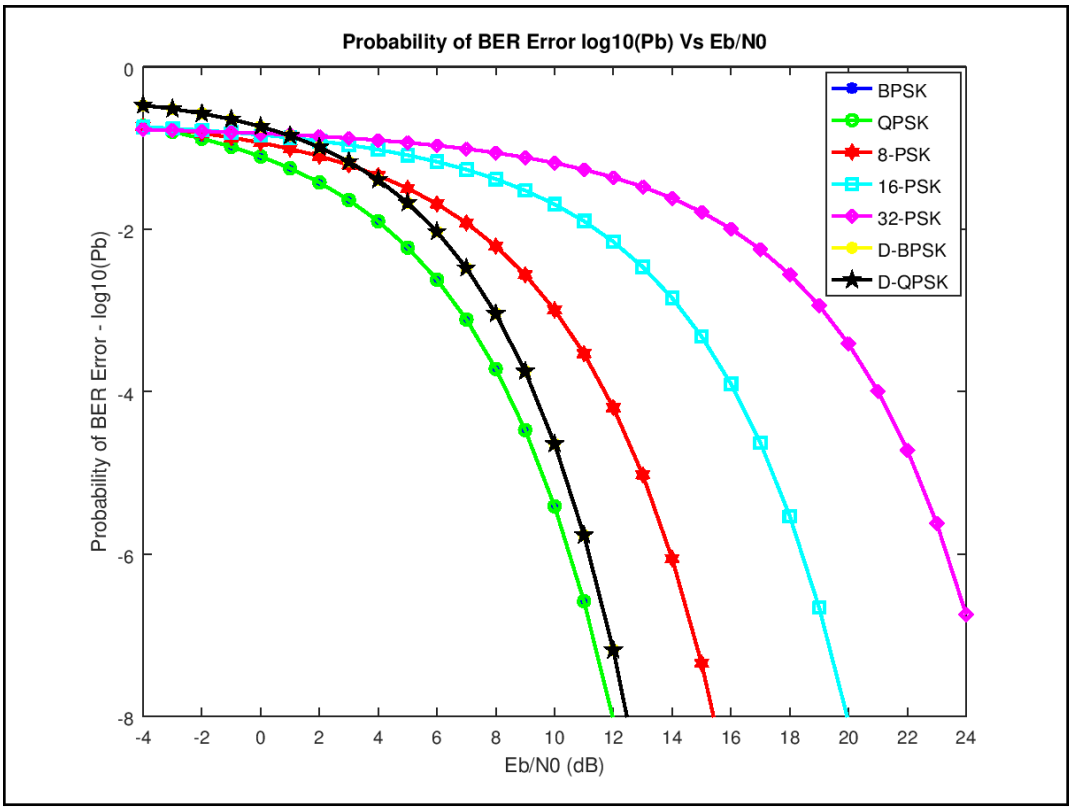
$$\frac{S}{N} = -15.528dB$$

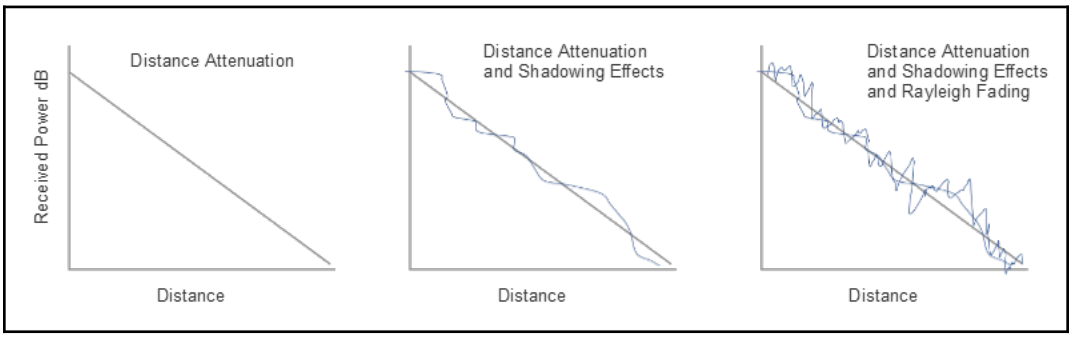
$$\frac{C}{B} = log_2(1 + \frac{E_b C}{N_0 B})$$

$$\frac{E_b}{N_0} = \frac{2^{\frac{C}{B}} - 1}{\frac{C}{B}}$$

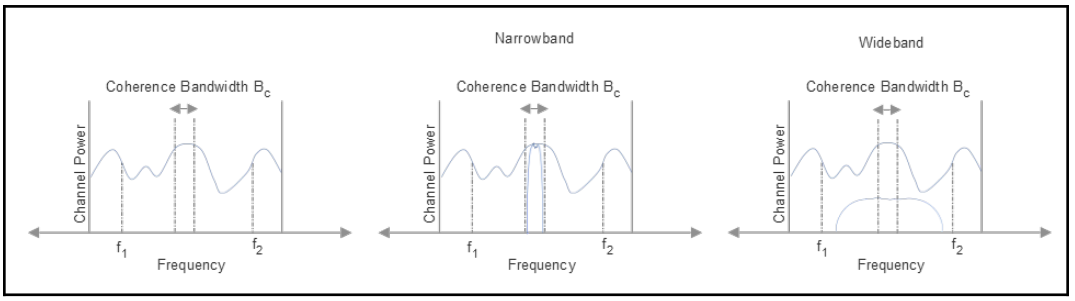
$$\frac{E_b}{N_0} \geq \lim_{\frac{C}{B} \to 0} \frac{2^{\frac{C}{B}} - 1}{\frac{C}{B}} = ln(2) = -1.59dB$$

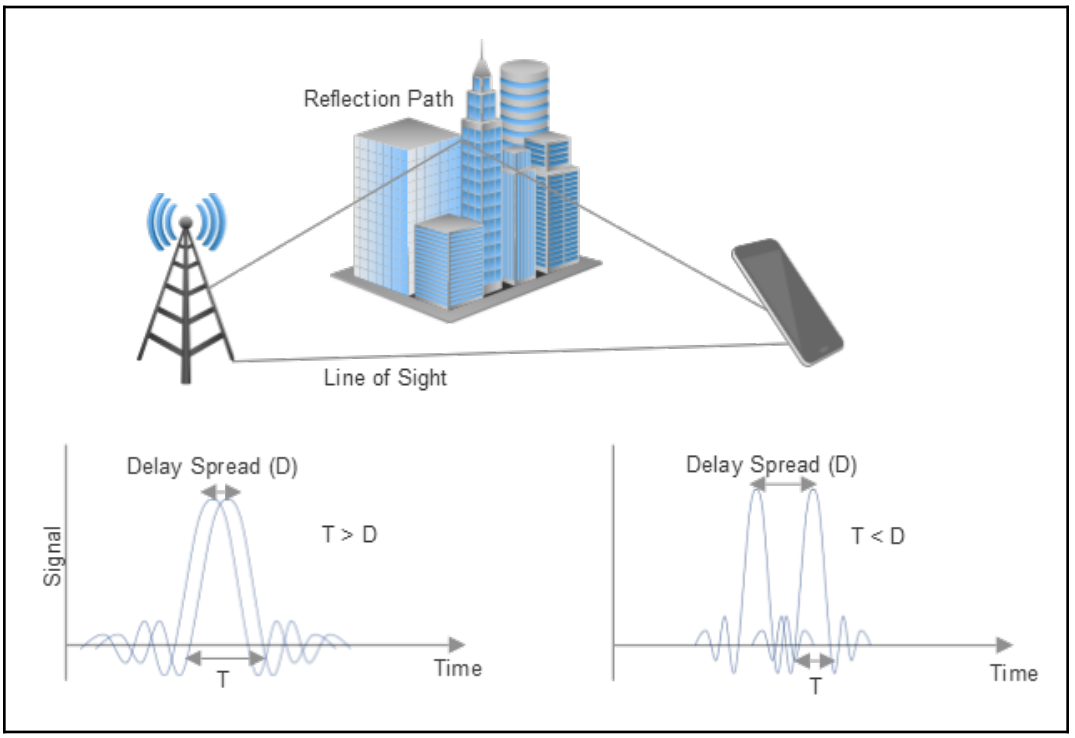

Spectral Efficiency vs Bit to Noise (SNR) Ratio
100
10
1
0.1
bits/S/Hz (R/B)
Shannon Limit ln(2) = -1.6
R>B Impossible Region
R=C Limit
Bandwidth Limited R/B > 1
64-QAM
4-PAM (SSB)
16-QAM
32-PSK
16-PSK
Typical 4G LTE
Operating Area
2-PAM (SSB BPSK)
4-QAM
QPSK
BPSK
R<B Realizable Region
Power Limited R/B < 1
-5
0
5
10
15
20
25
30
SNR per bit: E_b/N_0 dB



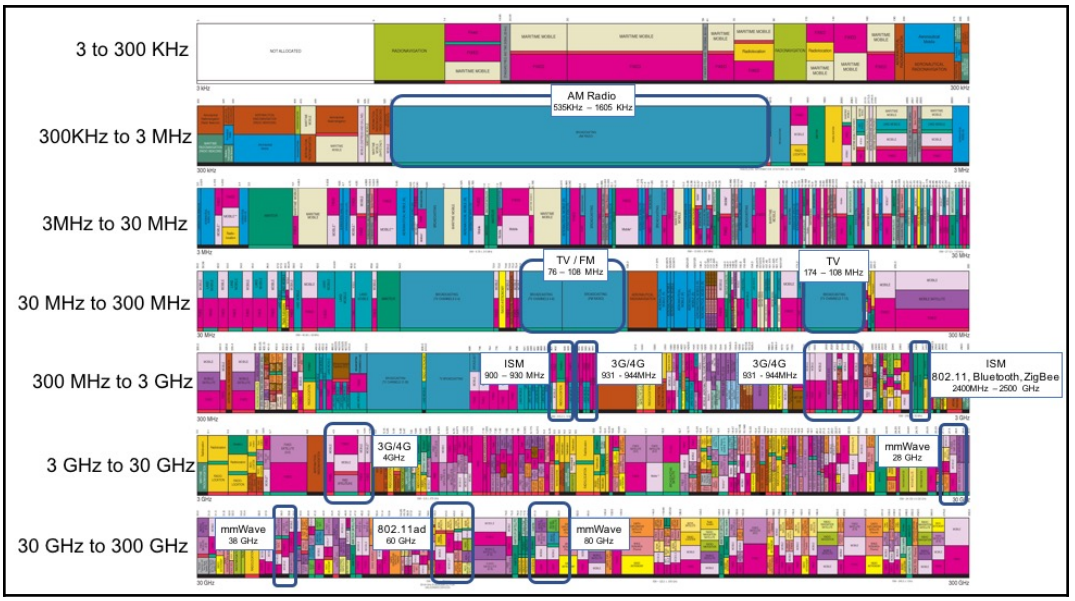


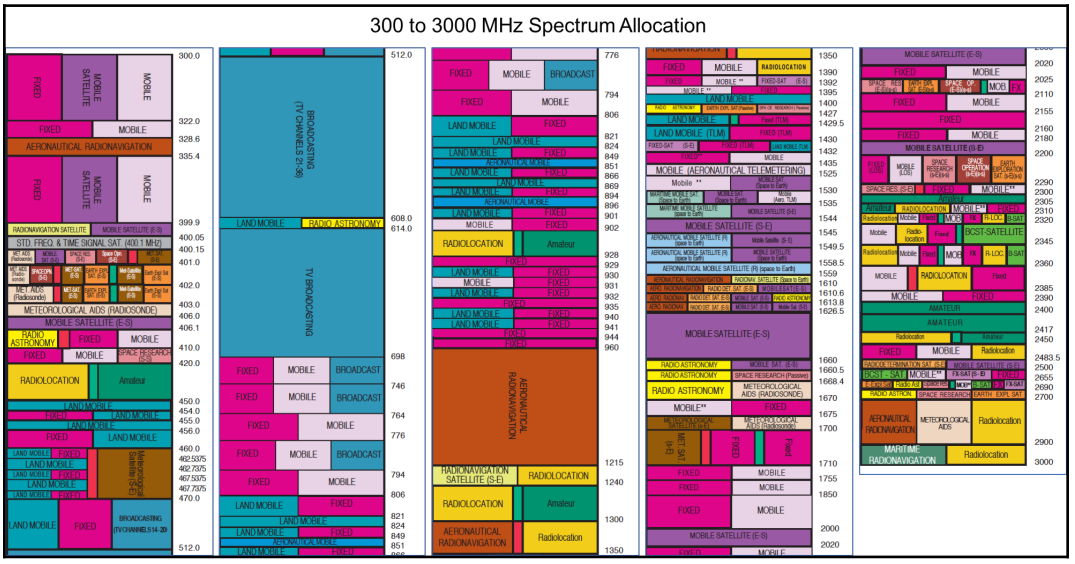
$$B_c \approx \frac{1}{D}$$


Narrowband
Wideband
Coherence Bandwidth B_c
Coherence Bandwidth B_c
Coherence Bandwidth B_c
Channel Power
Channel Power
Channel Power
f_1
f_2
Frequency
f_1
f_2
Frequency
f_1
f_2
Frequency

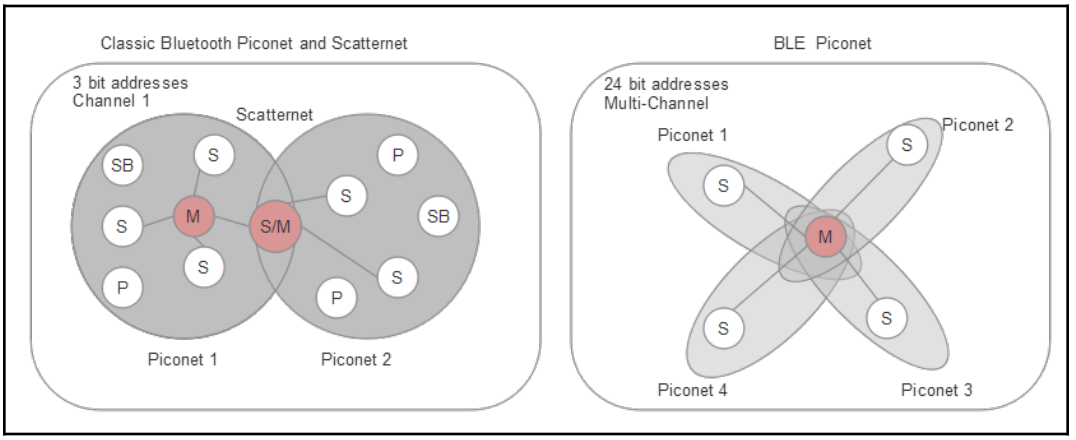

Reflection Path
Line of Sight
Delay Spread (D)
T > D
Signal
T
Time
Delay Spread (D)
T < D
T
Time

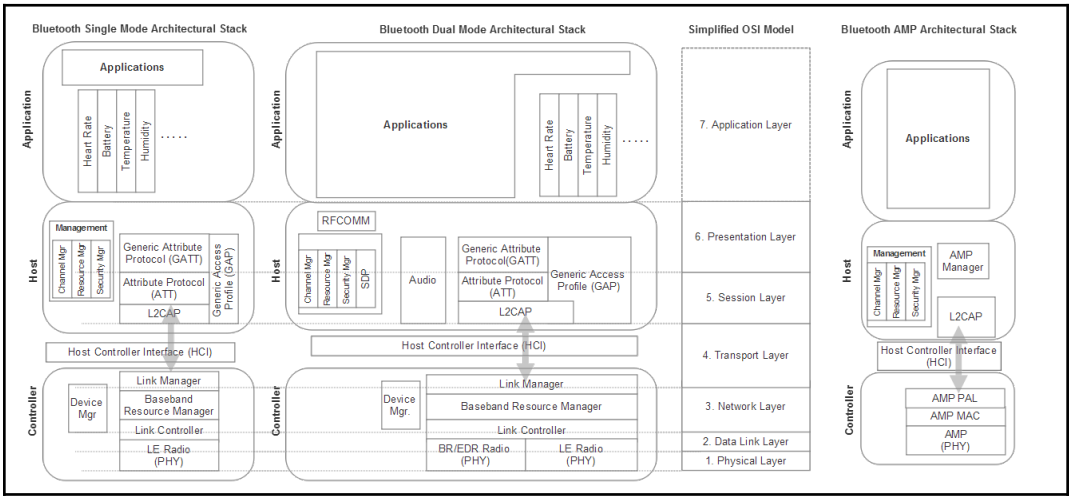
Frequency	IEEE Band	European Union, NATO, US ECM	ITU	
			ITU Band	ITU Abbreviation
0.3 Hz				
3 Hz		A	1	ELF
30 Hz			2	SLF
300 Hz			3	ULF
3 kHz			4	VLF
30 kHz			5	LF
300 kHz			6	MF
3 MHz	HF		7	HF
30 MHz	VHF		8	VHF
250 MHz		B		
300 MHz	UHF		9	UHF
500 MHz		C		
1 GHz	L	D		
2 GHz	S	E		
3 GHz		F	10	SHF
4 GHz	C	G		
6 GHz		H		
8 GHz	X	I		
10 GHz		J		
12 GHz	Ku			
18 GHz	K			
20 GHz		K		
27 GHz	Ka			
30 GHz			11	EHF
40 GHz	V	L		
60 GHz		M		
75 GHz	W			
100 GHz				
110 GHz	mm			
300 GHz			12	THF
3 THz				

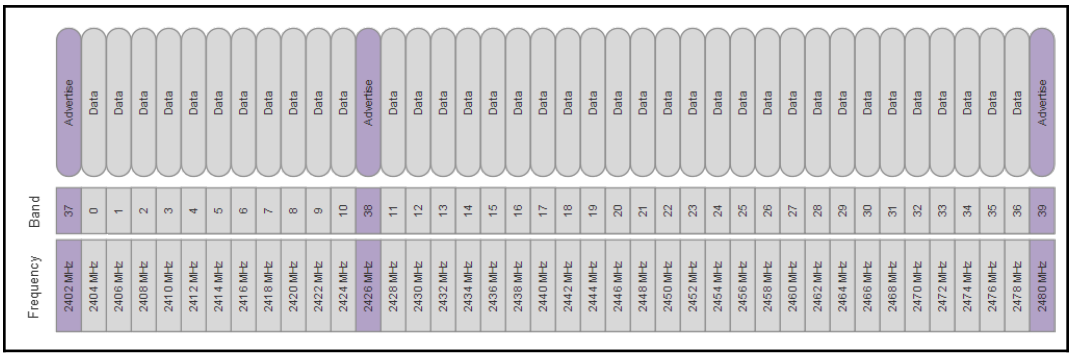

3 to 300 KHz
300KHz to 3 MHz
AM Radio
535Khz – 1605 KHz
3MHz to 30 MHz
30 MHz to 300 MHz
TV / FM
76 – 108 MHz
TV
174 – 108 MHz
300 MHz to 3 GHz
ISM
900 – 930 MHz
3G/4G
931 - 944MHz
3G/4G
931 - 944MHz
ISM
802.11, Bluetooth, ZigBee
2400MHz – 2500 GHz
3 GHz to 30 GHz
3G/4G
4GHz
mmWave
28 GHz
30 GHz to 300 GHz
mmWave
38 GHz
802.11ad
60 GHz
mmWave
80 GHz

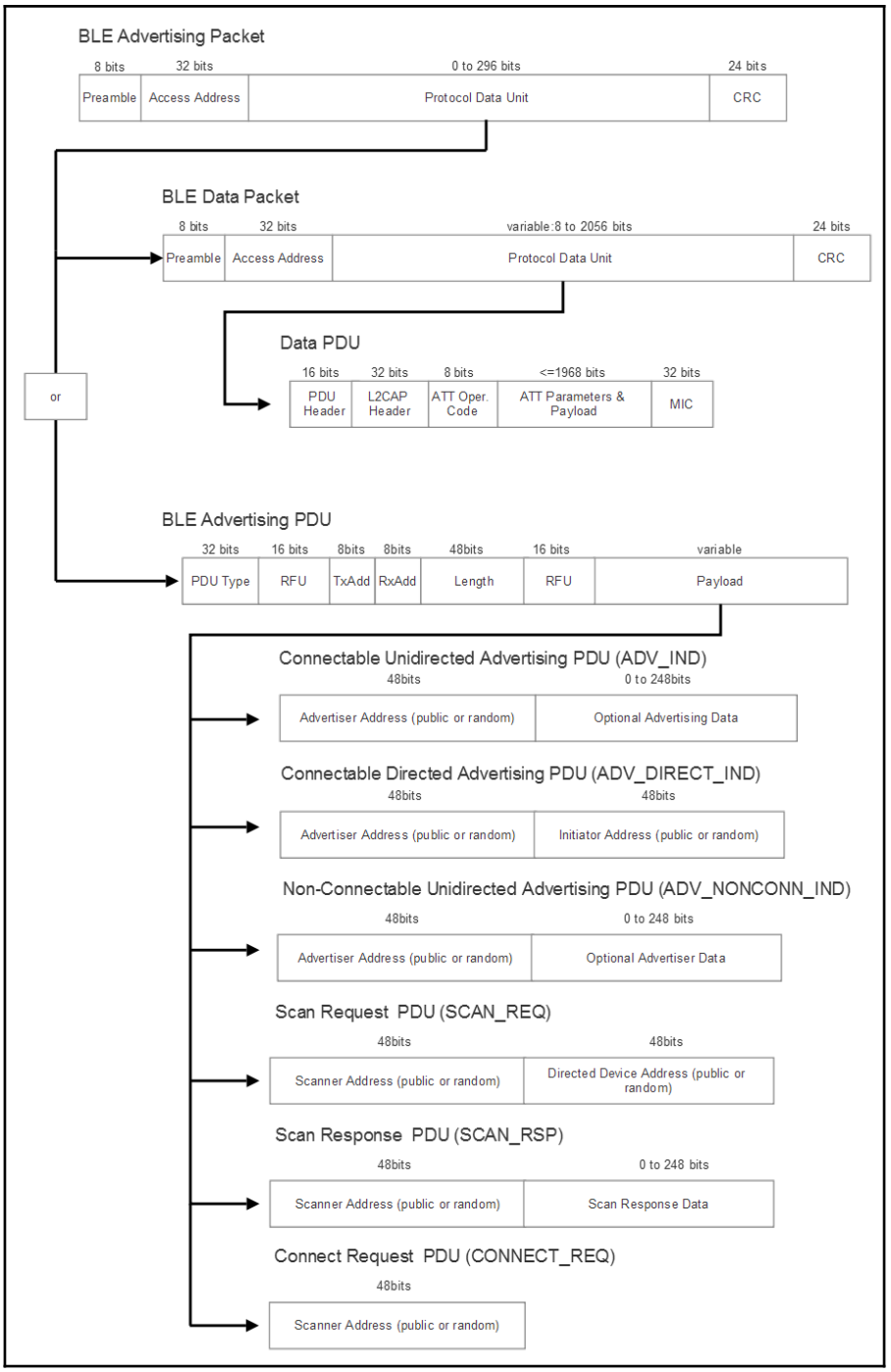

300 to 3000 MHz Spectrum Allocation

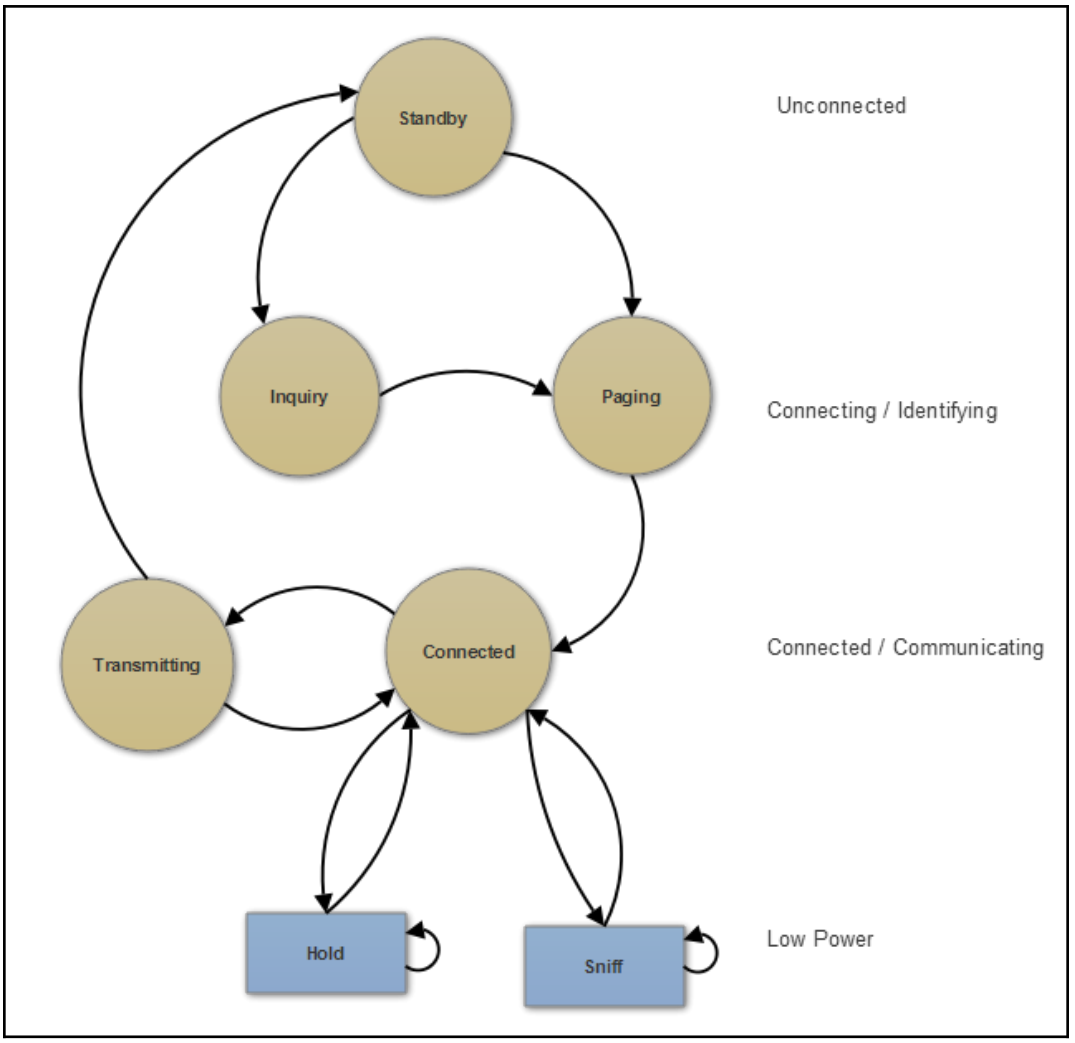
Chapter 5: Non-IP Based WPAN

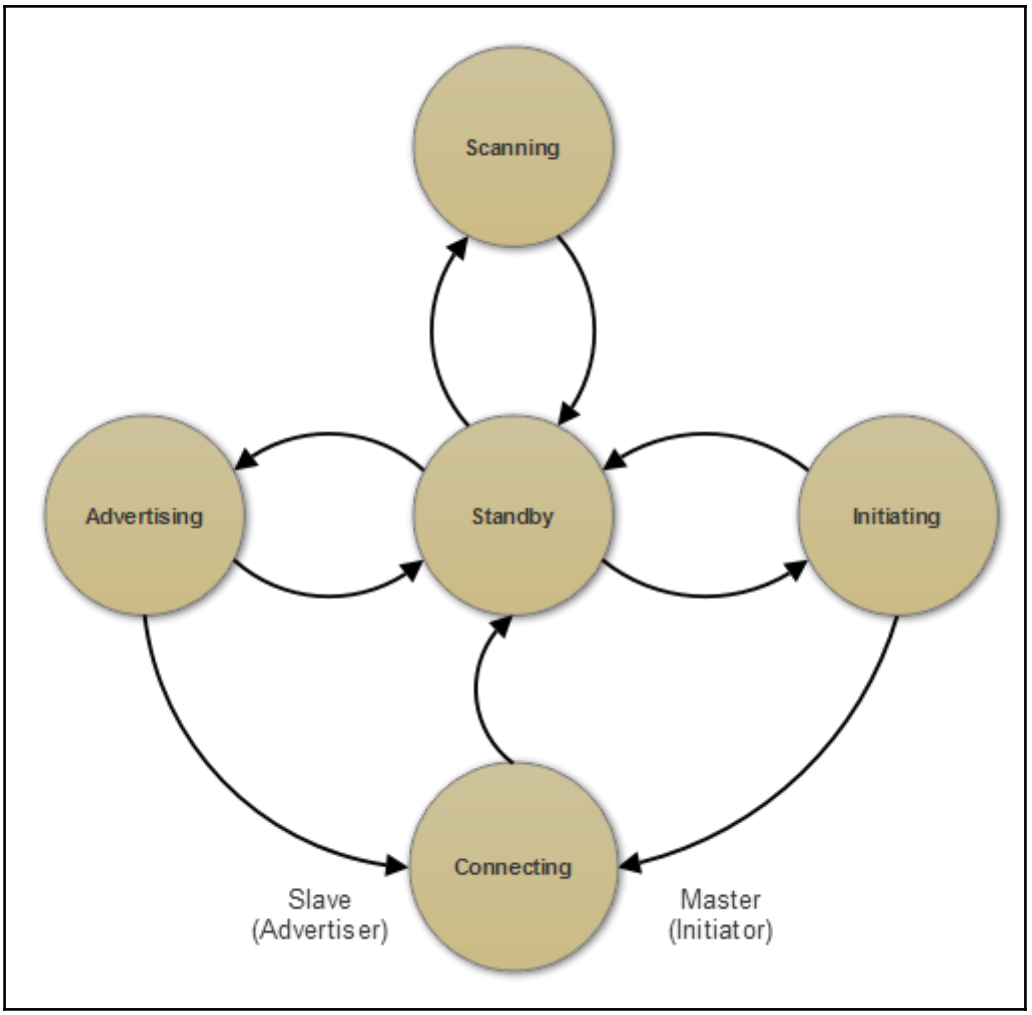


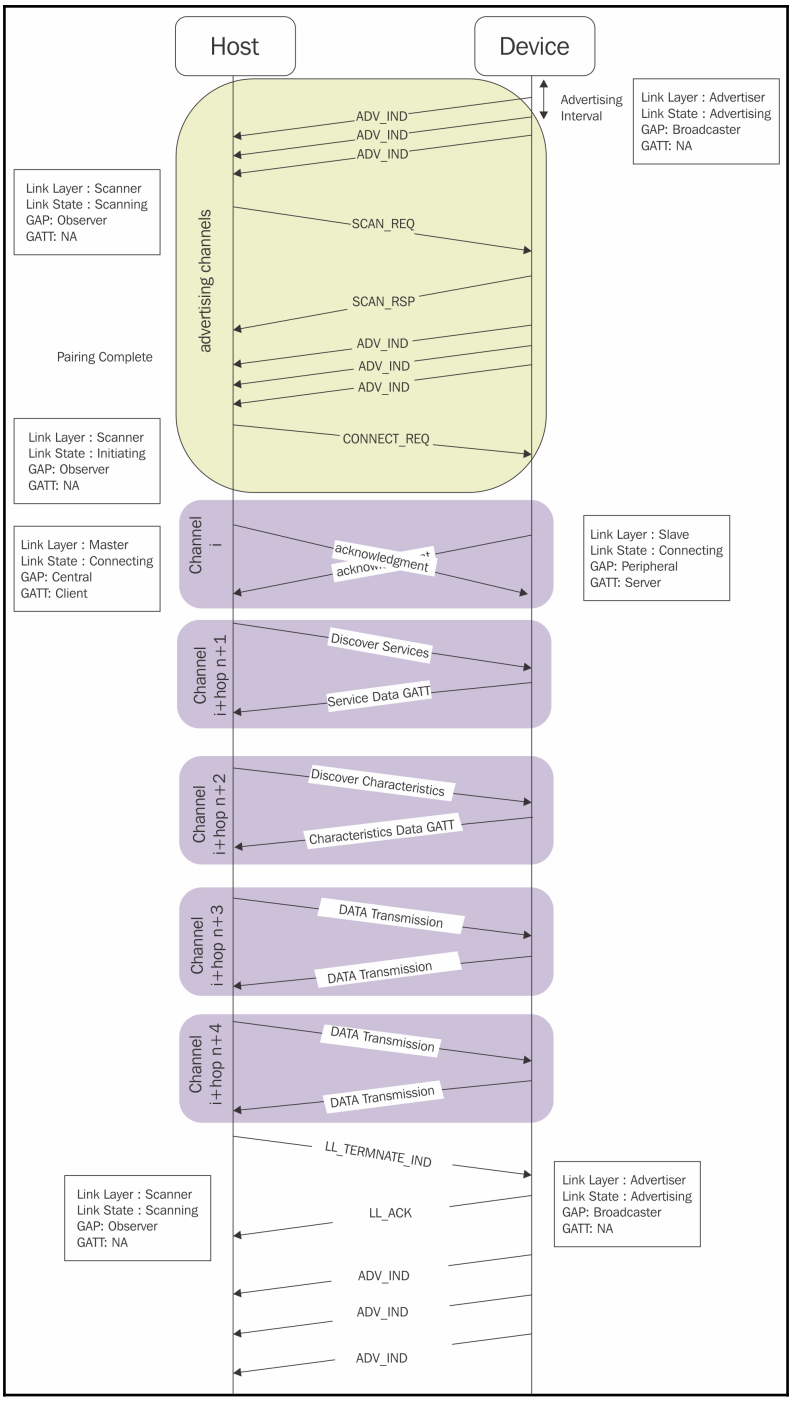


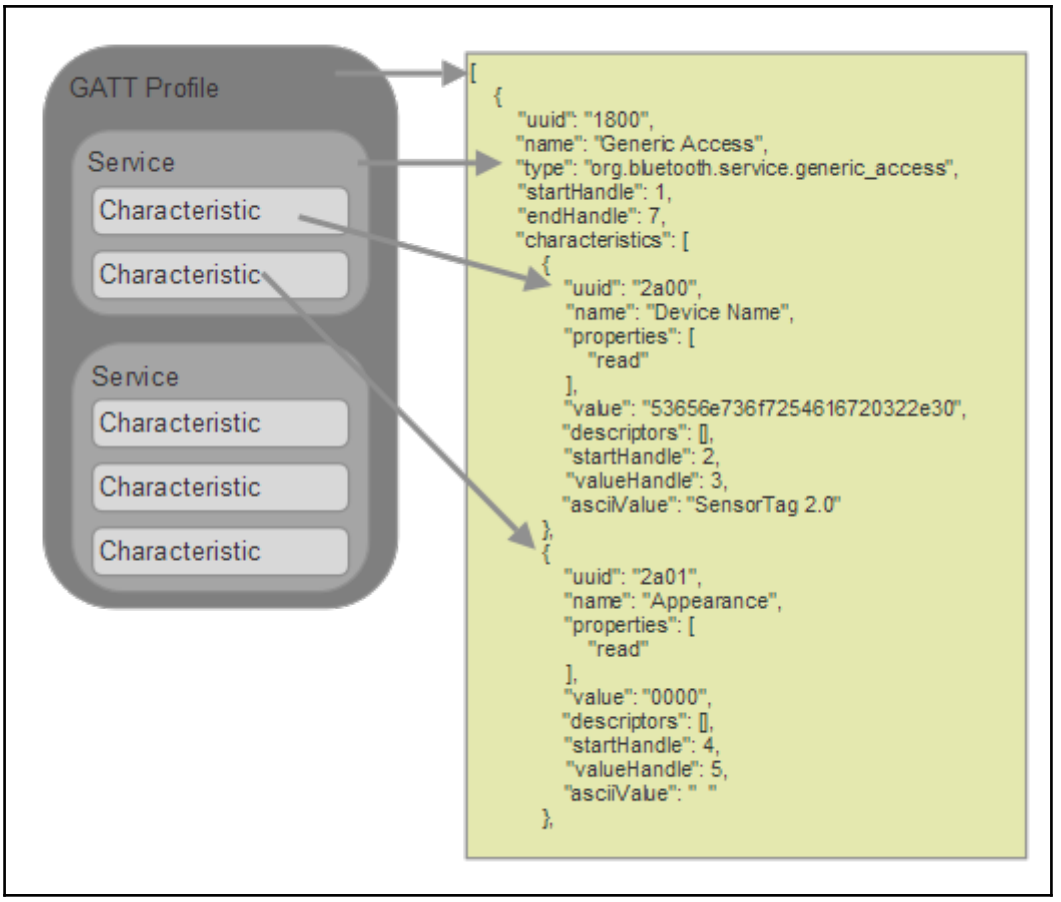

Frequency
Band
2402 MHz
37
Advertise
2404 MHz
0
Data
2406 MHz
1
Data
2408 MHz
2
Data
2410 MHz
3
Data
2412 MHz
4
Data
2414 MHz
5
Data
2416 MHz
6
Data
2418 MHz
7
Data
2420 MHz
8
Data
2422 MHz
9
Data
2424 MHz
10
Data
2426 MHz
38
Advertise
2428 MHz
11
Data
2430 MHz
12
Data
2432 MHz
13
Data
2434 MHz
14
Data
2436 MHz
15
Data
2438 MHz
16
Data
2440 MHz
17
Data
2442 MHz
18
Data
2444 MHz
19
Data
2446 MHz
20
Data
2448 MHz
21
Data
2450 MHz
22
Data
2452 MHz
23
Data
2454 MHz
24
Data
2456 MHz
25
Data
2458 MHz
26
Data
2460 MHz
27
Data
2462 MHz
28
Data
2464 MHz
29
Data
2466 MHz
30
Data
2468 MHz
31
Data
2470 MHz
32
Data
2472 MHz
33
Data
2474 MHz
34
Data
2476 MHz
35
Data
2478 MHz
36
Data
2480 MHz
39
Advertise

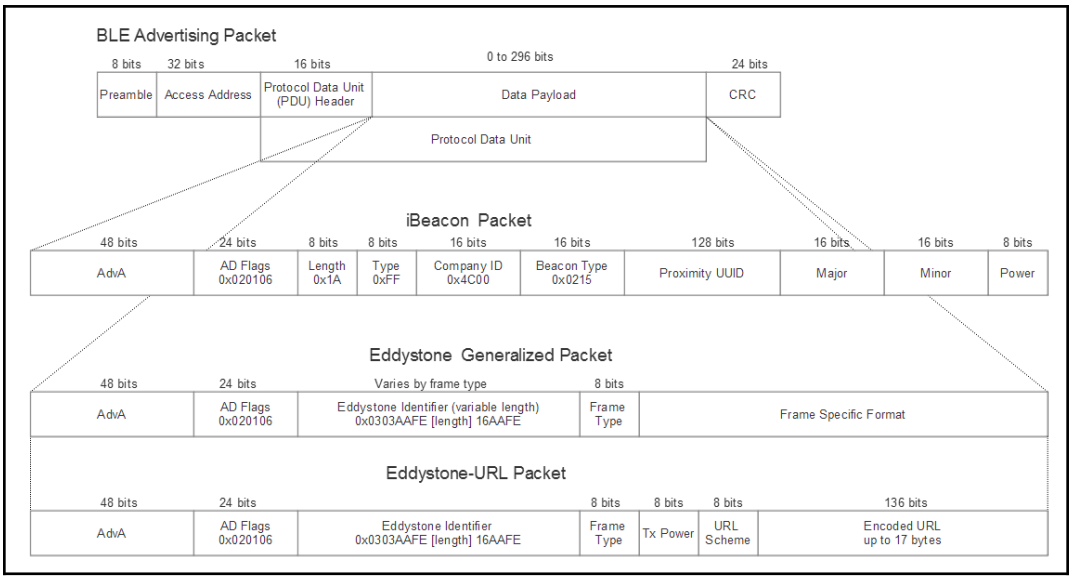

BLE Advertising Packet
8 bits
32 bits
0 to 296 bits
24 bits
Preamble
Access Address
Protocol Data Unit
CRC
BLE Data Packet
8 bits
32 bits
variable:8 to 2056 bits
24 bits
Preamble
Access Address
Protocol Data Unit
CRC
Data PDU
16 bits
32 bits
8 bits
<=1968 bits
32 bits
PDU Header
L2CAP Header
ATT Oper. Code
ATT Parameters & Payload
MIC
or
BLE Advertising PDU
32 bits
16 bits
8bits
8bits
48bits
16 bits
variable
PDU Type
RFU
TxAdd
RxAdd
Length
RFU
Payload
Connectable Unidirected Advertising PDU (ADV_IND)
48bits
0 to 248bits
Advertiser Address (public or random)
Optional Advertising Data
Connectable Directed Advertising PDU (ADV_DIRECT_IND)
48bits
48bits
Advertiser Address (public or random)
Initiator Address (public or random)
Non-Connectable Unidirected Advertising PDU (ADV_NONCONN_IND)
48bits
0 to 248 bits
Advertiser Address (public or random)
Optional Advertiser Data
Scan Request PDU (SCAN_REQ)
48bits
48bits
Scanner Address (public or random)
Directed Device Address (public or random)
Scan Response PDU (SCAN_RSP)
48bits
0 to 248 bits
Scanner Address (public or random)
Scan Response Data
Connect Request PDU (CONNECT_REQ)
48bits
Scanner Address (public or random)

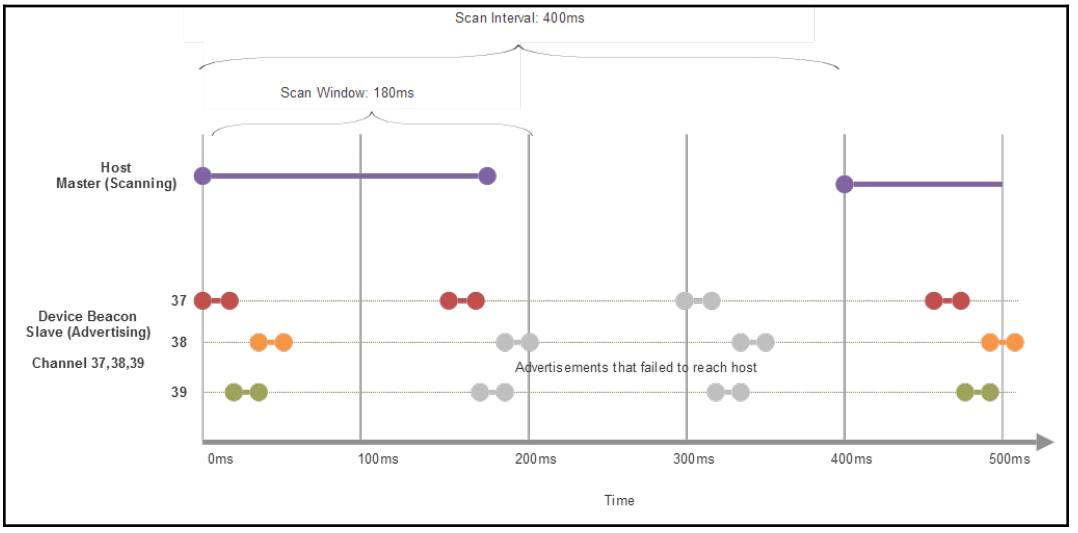

Standby
Unconnected
Inquiry
Paging
Connecting / Identifying
Transmitting
Connected
Connected / Communicating
Hold
Sniff
Low Power

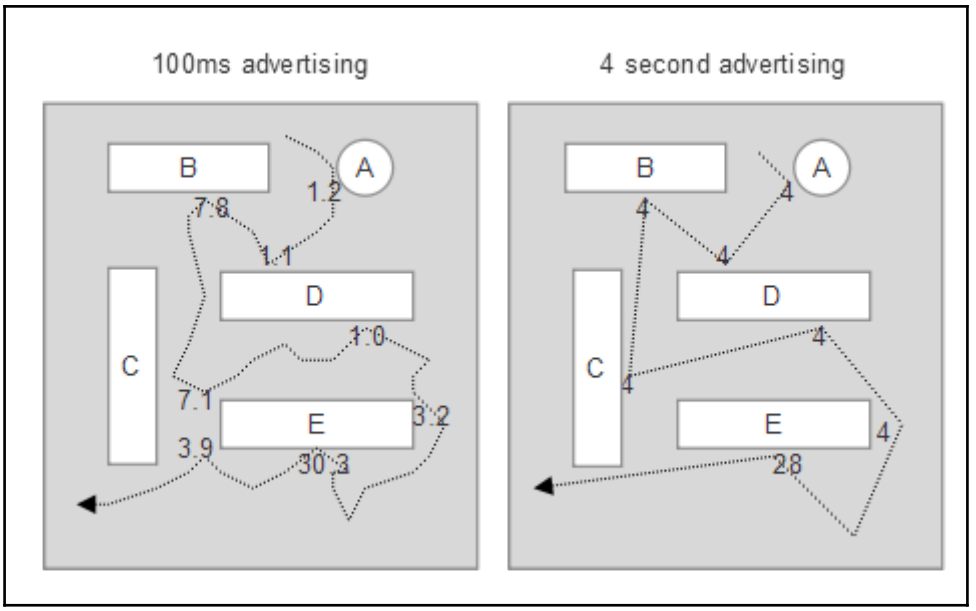

Scanning
Advertising
Standby
Initiating
Connecting
Slave
(Advertiser)
Master
(Initiator)

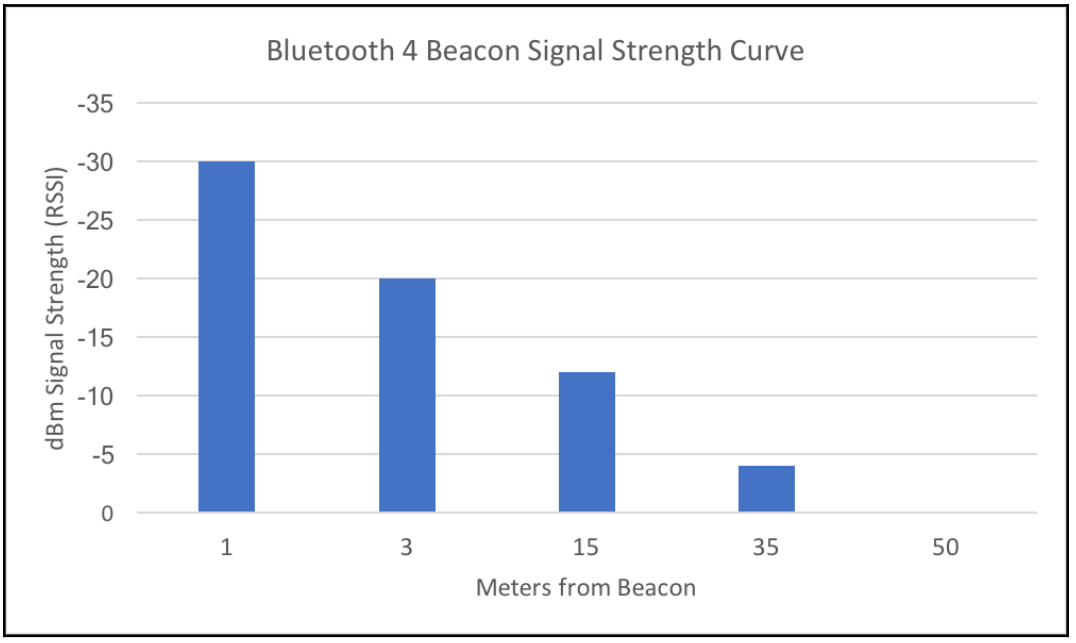

Host
Device
Advertising Interval
Link Layer : Advertiser
Link State : Advertising
GAP: Broadcaster
GATT: NA
ADV_IND
ADV_IND
ADV_IND
Link Layer : Scanner
Link State : Scanning
GAP: Observer
GATT: NA
SCAN_REQ
SCAN_RSP
advertising channels
Pairing Complete
ADV_IND
ADV_IND
ADV_IND
Link Layer : Scanner
Link State : Initiating
GAP: Observer
GATT: NA
CONNECT_REQ
Link Layer : Master
Link State : Connecting
GAP: Central
GATT: Client
Channel i
Link Layer : Slave
Link State : Connecting
GAP: Peripheral
GATT: Server
Channel i+hop n+1
Discover Services
Service Data GATT
Channel i+hop n+2
Discover Characteristics
Characteristics Data GATT
Channel i+hop n+3
DATA Transmission
DATA Transmission
Channel i+hop n+4
DATA Transmission
DATA Transmission
LL_TERMINATE_IND
Link Layer : Scanner
Link State : Scanning
GAP: Observer
GATT: NA
LL_ACK
Link Layer : Advertiser
Link State : Advertising
GAP: Broadcaster
GATT: NA
ADV_IND
ADV_IND
ADV_IND

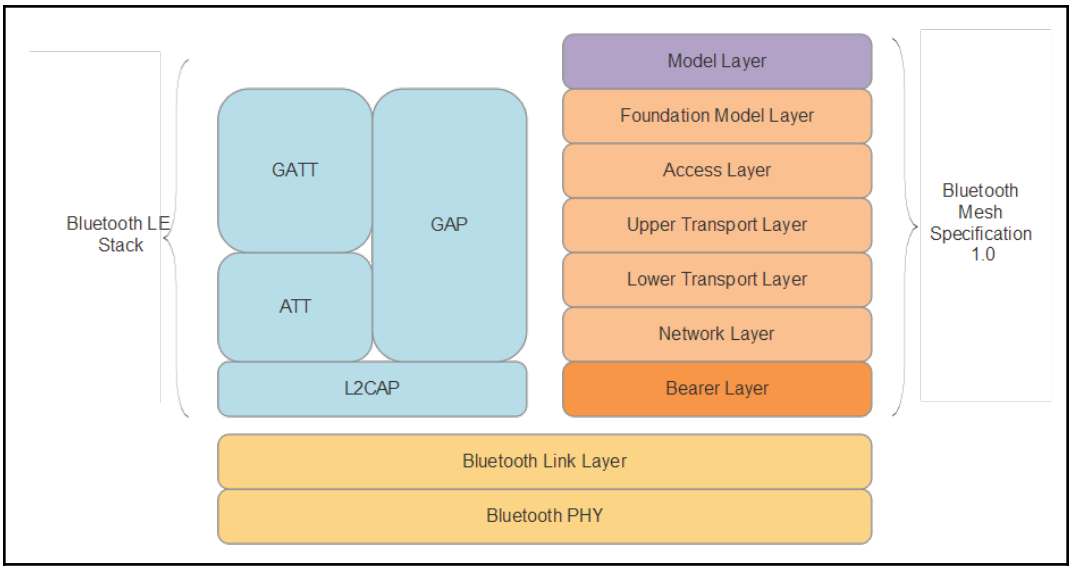

GATT Profile
Service
Characteristic
Characteristic
Service
Characteristic
Characteristic
Characteristic
[
{
"uuid": "1800",
"name": "Generic Access",
"type": "org.bluetooth.service.generic_access",
"startHandle": 1,
"endHandle": 7,
"characteristics": [
{
"uuid": "2a00",
"name": "Device Name",
"properties": [
"read"
],
"value": "53656e736f7254616720322e30",
"descriptors": [],
"startHandle": 2,
"valueHandle": 3,
"asciiValue": "SensorTag 2.0"
},
{
"uuid": "2a01",
"name": "Appearance",
"properties": [
"read"
],
"value": "0000",
"descriptors": [],
"startHandle": 4,
"valueHandle": 5,
"asciiValue": " "
},

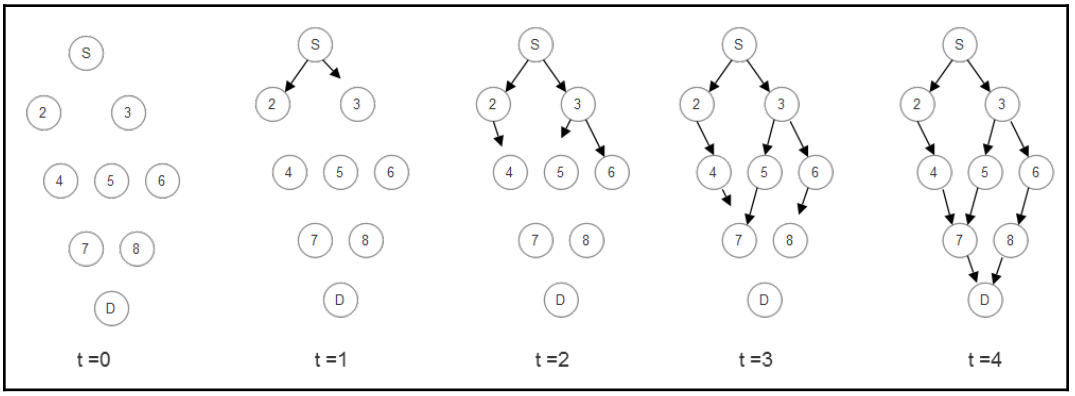

BLE Advertising Packet
8 bits
32 bits
16 bits
0 to 296 bits
24 bits
Preamble
Access Address
Protocol Data Unit (PDU) Header
Data Payload
CRC
Protocol Data Unit
iBeacon Packet
48 bits
24 bits
8 bits
8 bits
16 bits
16 bits
128 bits
16 bits
16 bits
8 bits
AdvA
AD Flags 0x020106
Length 0x1A
Type 0xFF
Company ID 0x4C00
Beacon Type 0x0215
Proximity UUID
Major
Minor
Power
Eddystone Generalized Packet
48 bits
24 bits
Varies by frame type
8 bits
AdvA
AD Flags 0x020106
Eddystone Identifier (variable length) 0x0303AAFE [length] 16AAFE
Frame Type
Frame Specific Format
Eddystone-URL Packet
48 bits
24 bits
8 bits
8 bits
8 bits
136 bits
AdvA
AD Flags 0x020106
Eddystone Identifier 0x0303AAFE [length] 16AAFE
Frame Type
Tx Power
URL Scheme
Encoded URL up to 17 bytes

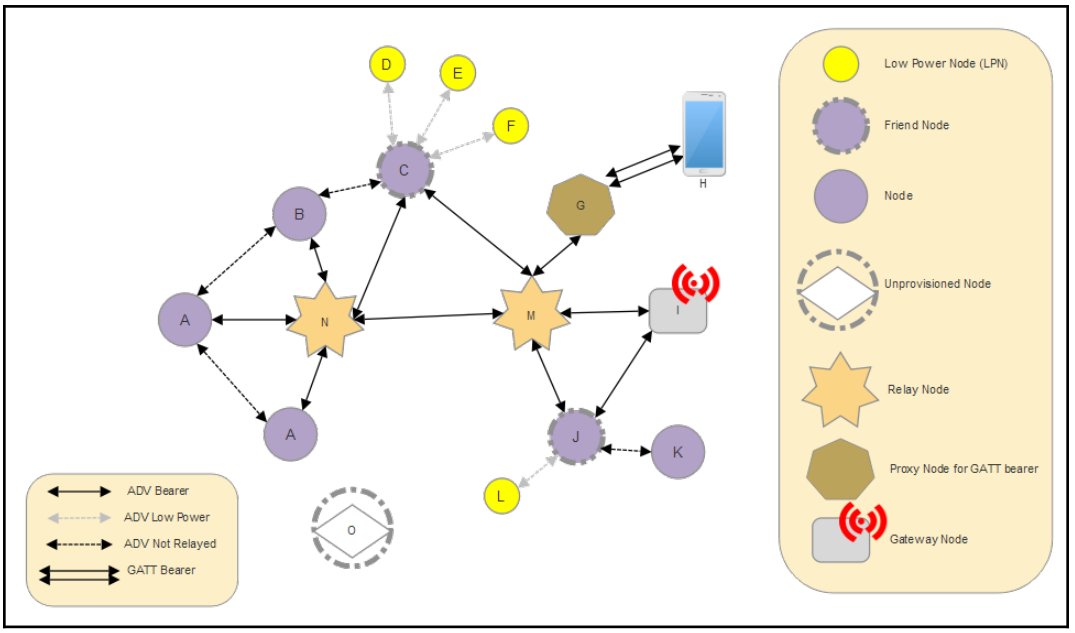

Scan Interval: 400ms
Scan Window: 180ms
Host
Master (Scanning)
Device Beacon
Slave (Advertising)
Channel 37,38,39
37
38
39
Advertisements that failed to reach host
0ms
100ms
200ms
300ms
400ms
500ms
Time

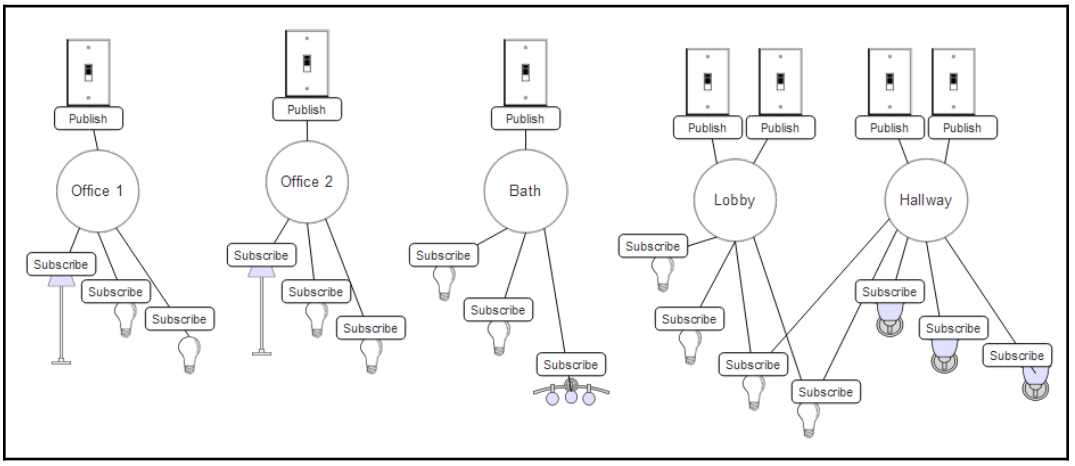

100ms advertising
4 second advertising
B
A
C
D
E
1.2
7.8
1.1
1.0
7.1
3.2
3.9
30.3
4
4
4
4
4
4
2.8

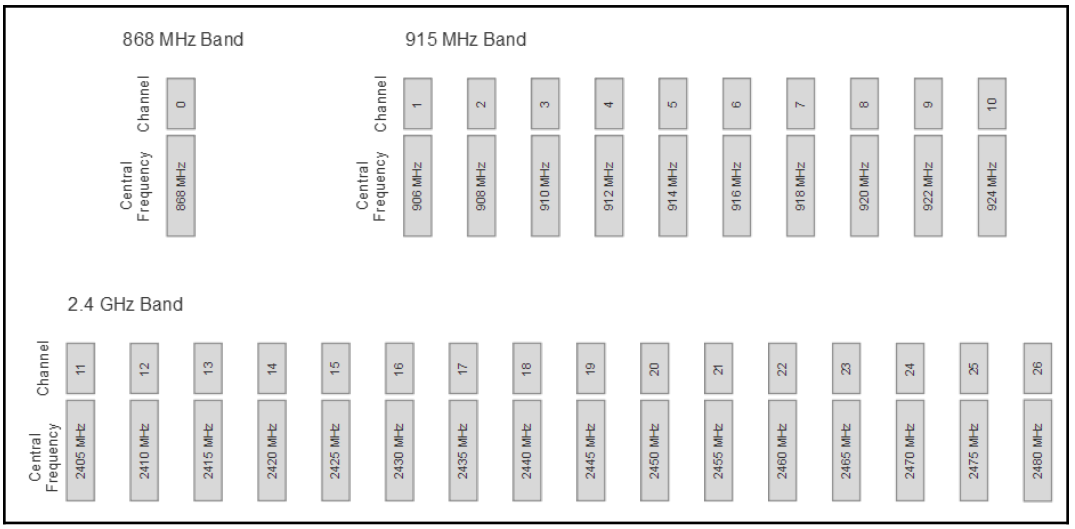

Bluetooth 4 Beacon Signal Strength Curve
dBm Signal Strength (RSSI)
-35
-30
-25
-20
-15
-10
-5
0
1
3
15
35
50
Meters from Beacon

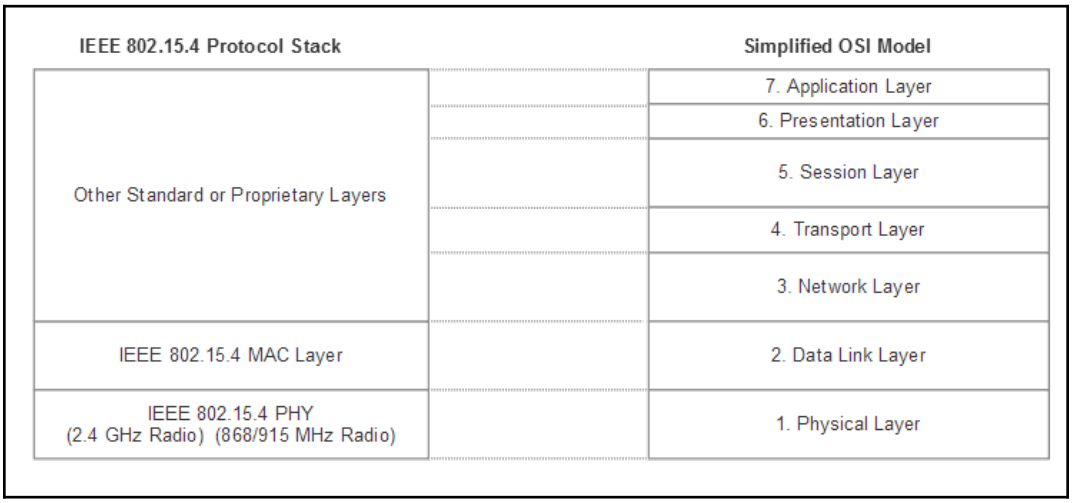

Bluetooth LE Stack
GATT
GAP
ATT
L2CAP
Model Layer
Foundation Model Layer
Access Layer
Upper Transport Layer
Lower Transport Layer
Network Layer
Bearer Layer
Bluetooth Mesh Specification 1.0
Bluetooth Link Layer
Bluetooth PHY

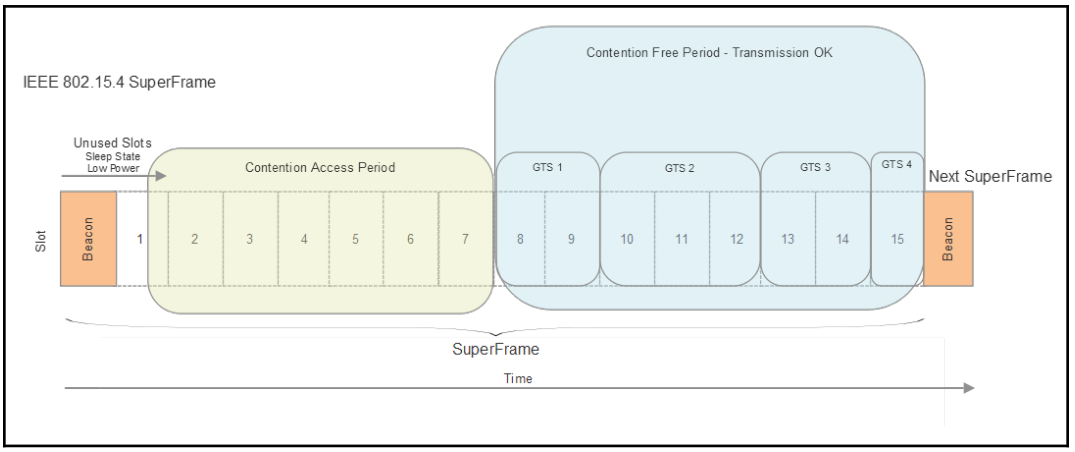

S
2
3
4
5
6
7
8
D
t =0
t =1
t =2
t =3
t =4

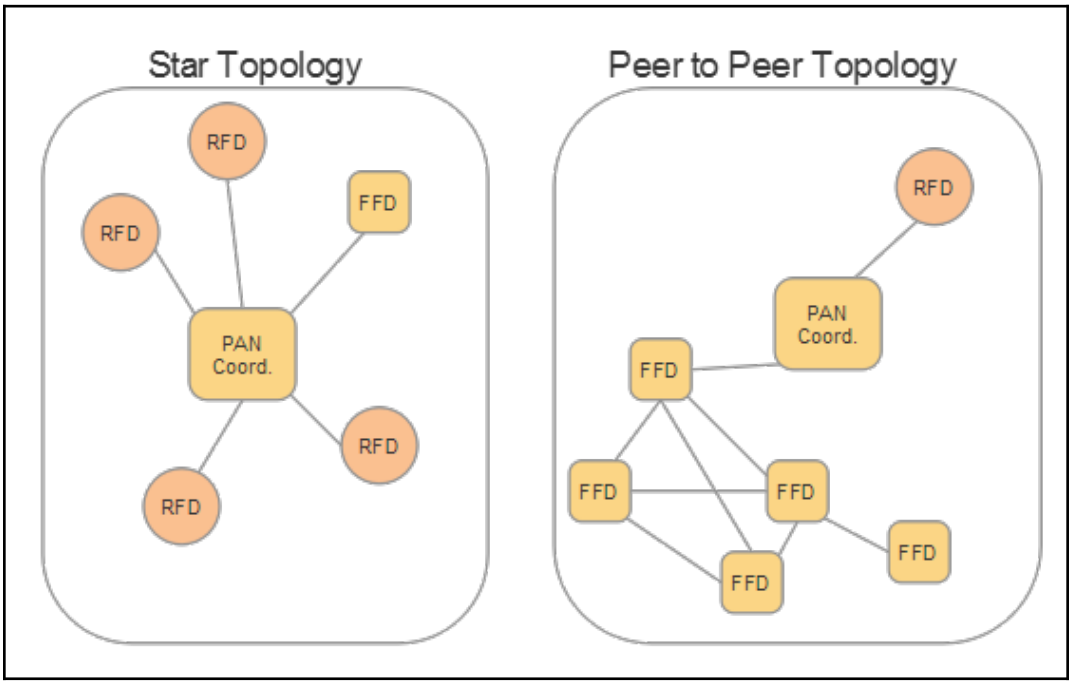

D
E
F
C
B
G
H
A
N
M
I
A
J
K
L
O
ADV Bearer
ADV Low Power
ADV Not Relayed
GATT Bearer
Low Power Node (LPN)
Friend Node
Node
Unprovisioned Node
Relay Node
Proxy Node for GATT bearer
Gateway Node

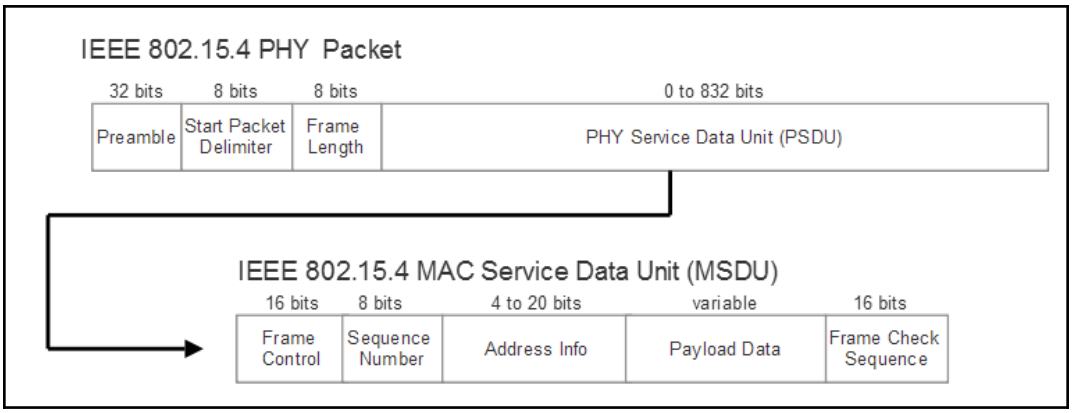

Publish
Office 1
Subscribe
Subscribe
Subscribe
Publish
Office 2
Subscribe
Subscribe
Subscribe
Publish
Bath
Subscribe
Subscribe
Subscribe
Publish
Publish
Lobby
Subscribe
Subscribe
Subscribe
Subscribe
Publish
Publish
Hallway
Subscribe
Subscribe
Subscribe

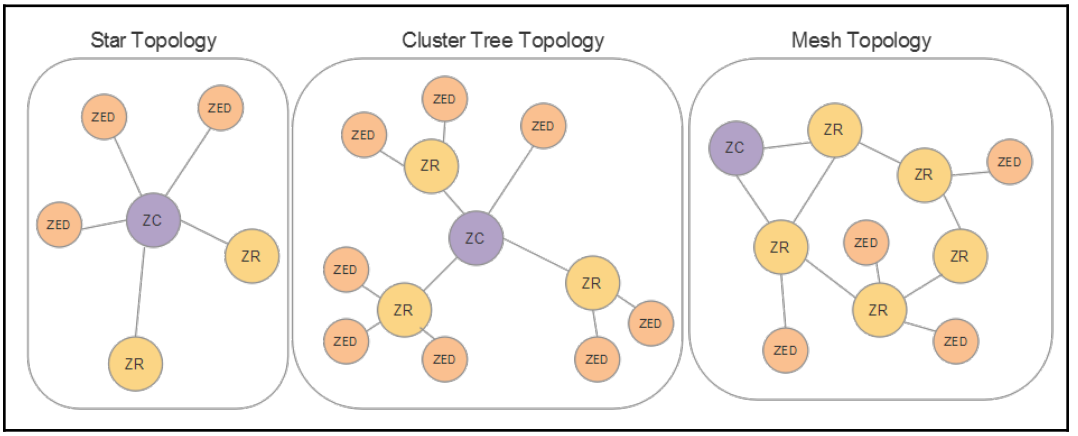

868 MHz Band
915 MHz Band
Channel
Central Frequency
0
868 MHz
1
906 MHz
2
908 MHz
3
910 MHz
4
912 MHz
5
914 MHz
6
916 MHz
7
918 MHz
8
920 MHz
9
922 MHz
10
924 MHz
2.4 GHz Band
11
2405 MHz
12
2410 MHz
13
2415 MHz
14
2420 MHz
15
2425 MHz
16
2430 MHz
17
2435 MHz
18
2440 MHz
19
2445 MHz
20
2450 MHz
21
2455 MHz
22
2460 MHz
23
2465 MHz
24
2470 MHz
25
2475 MHz
26
2480 MHz

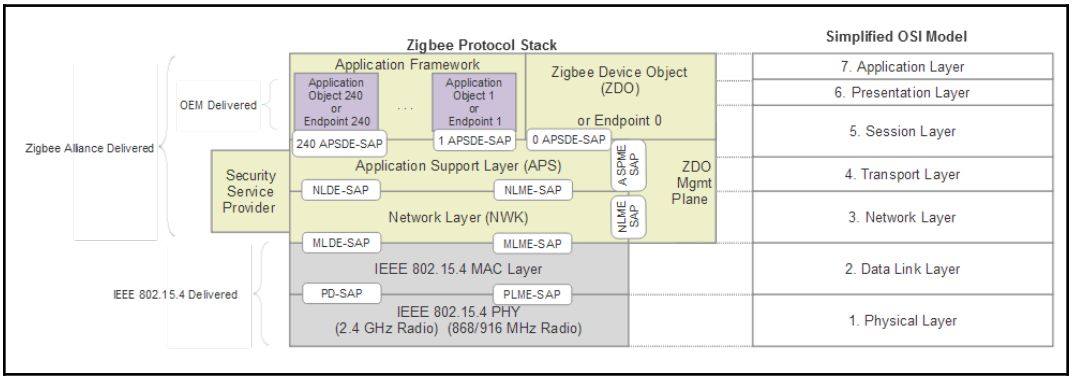

IEEE 802.15.4 Protocol Stack
Simplified OSI Model
Other Standard or Proprietary Layers
IEEE 802.15.4 MAC Layer
IEEE 802.15.4 PHY
(2.4 GHz Radio) (868/915 MHz Radio)
7. Application Layer
6. Presentation Layer
5. Session Layer
4. Transport Layer
3. Network Layer
2. Data Link Layer
1. Physical Layer

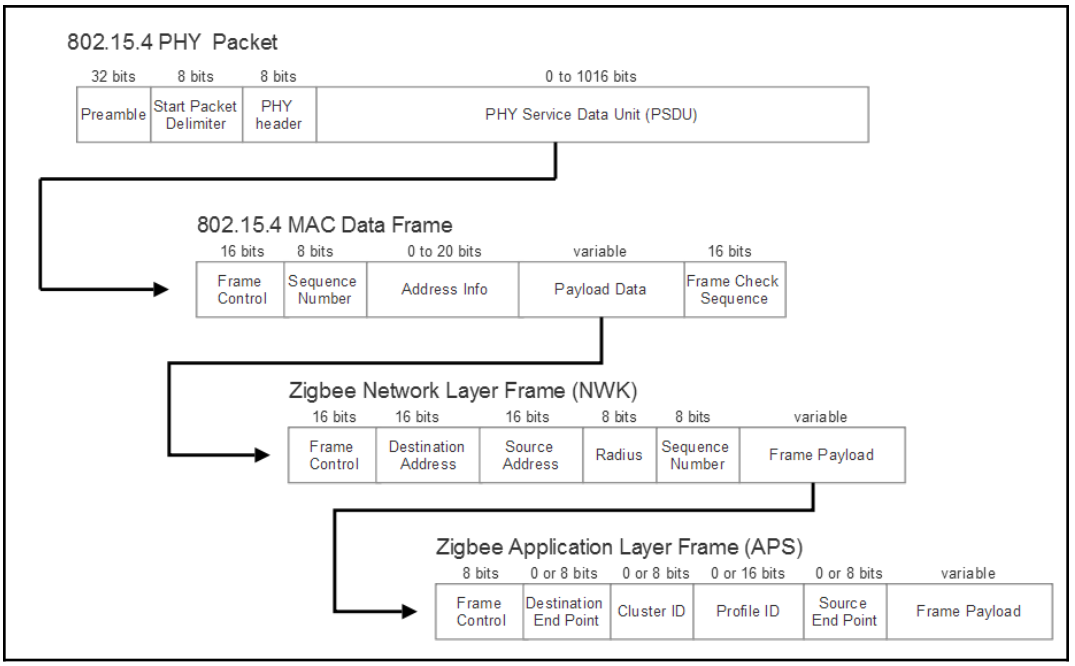

IEEE 802.15.4 SuperFrame
Contention Free Period - Transmission OK
Unused Slots
Sleep State
Low Power
Contention Access Period
GTS 1
GTS 2
GTS 3
GTS 4
Next SuperFrame
Slot
Beacon
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Beacon
SuperFrame
Time

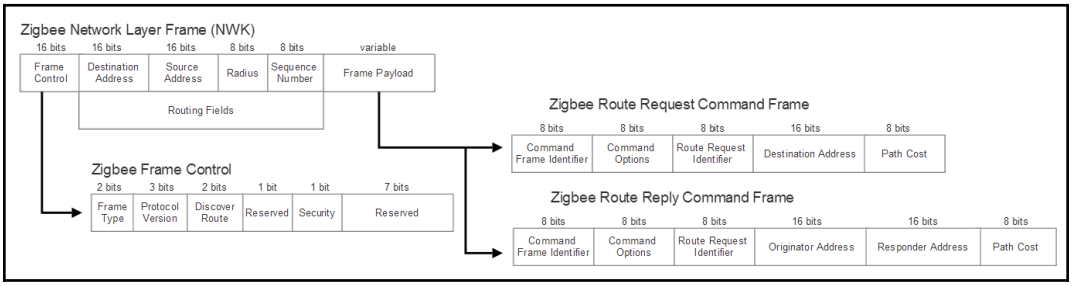

Star Topology
RFD
FFD
RFD
PAN
Coord.
RFD
RFD
Peer to Peer Topology
RFD
PAN
Coord.
FFD
FFD
FFD
FFD
FFD

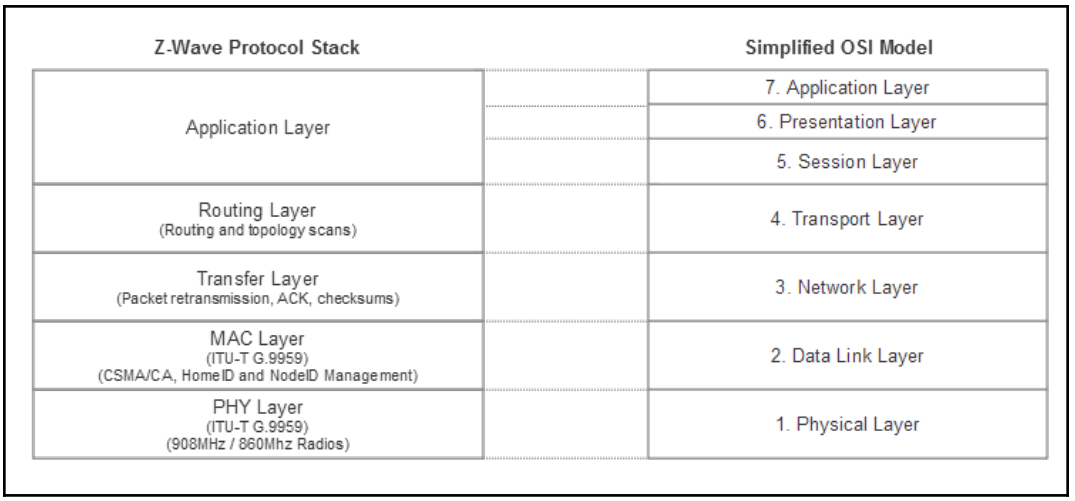

IEEE 802.15.4 PHY Packet
32 bits
8 bits
8 bits
0 to 832 bits
Preamble
Start Packet Delimiter
Frame Length
PHY Service Data Unit (PSDU)
IEEE 802.15.4 MAC Service Data Unit (MSDU)
16 bits
8 bits
4 to 20 bits
variable
16 bits
Frame Control
Sequence Number
Address Info
Payload Data
Frame Check Sequence

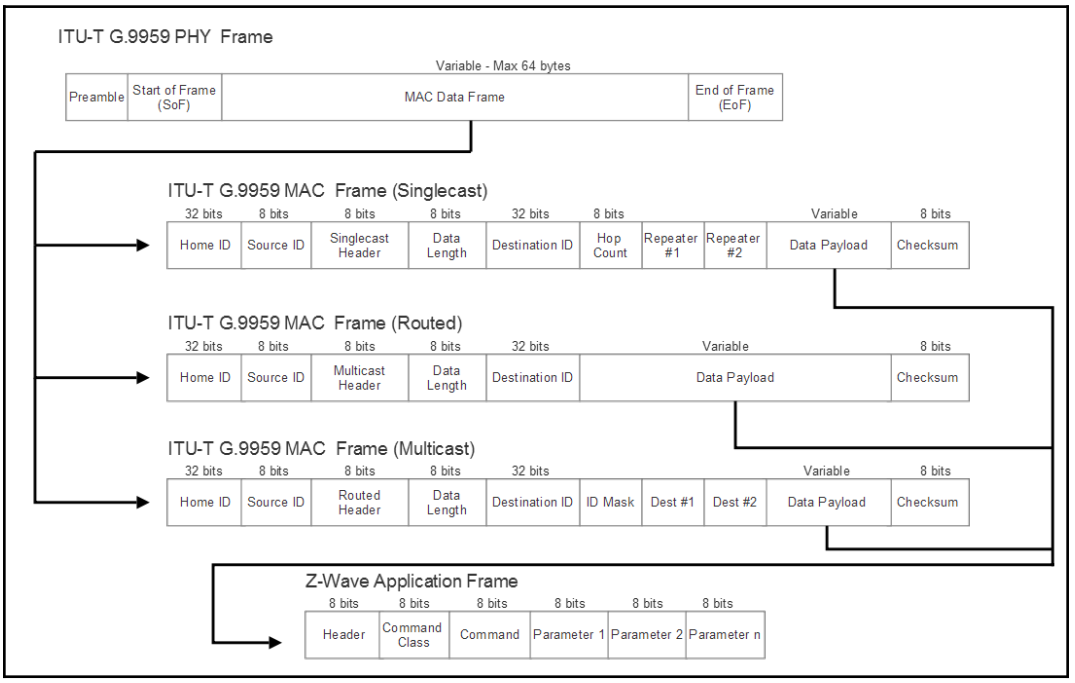

Star Topology
Cluster Tree Topology
Mesh Topology
ZC
ZR
ZED

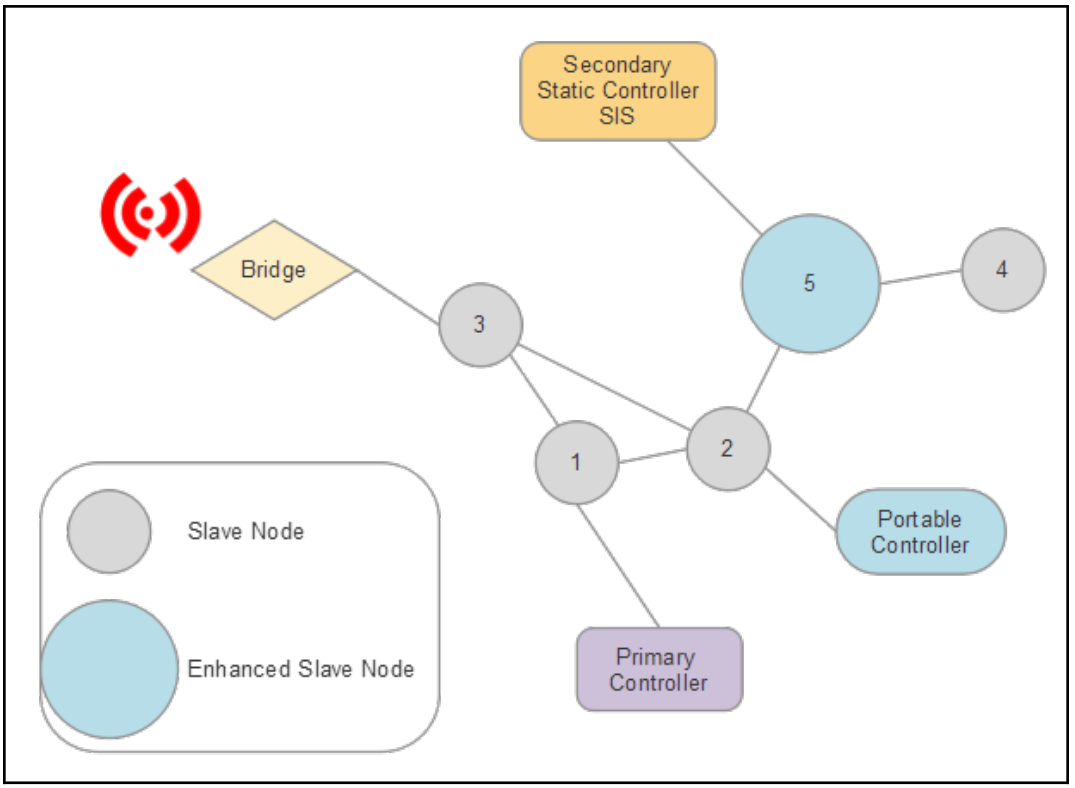

Zigbee Protocol Stack
Simplified OSI Model
Application Framework
Application Object 240 or Endpoint 240
Application Object 1 or Endpoint 1
Zigbee Device Object (ZDO)
or Endpoint 0
240 APSDE-SAP
1 APSDE-SAP
0 APSDE-SAP
Application Support Layer (APS)
ASPME SAP
NLME SAP
NLDE-SAP
NLME-SAP
Network Layer (NWK)
MLDE-SAP
MLME-SAP
IEEE 802.15.4 MAC Layer
PD-SAP
PLME-SAP
IEEE 802.15.4 PHY
(2.4 GHz Radio) (868/916 MHz Radio)
OEM Delivered
Zigbee Alliance Delivered
Security Service Provider
IEEE 802.15.4 Delivered
ZDO Mgmt Plane
7. Application Layer
6. Presentation Layer
5. Session Layer
4. Transport Layer
3. Network Layer
2. Data Link Layer
1. Physical Layer


802.15.4 PHY Packet
32 bits
8 bits
8 bits
0 to 1016 bits
Preamble
Start Packet Delimiter
PHY header
PHY Service Data Unit (PSDU)
802.15.4 MAC Data Frame
16 bits
8 bits
0 to 20 bits
variable
16 bits
Frame Control
Sequence Number
Address Info
Payload Data
Frame Check Sequence
Zigbee Network Layer Frame (NWK)
16 bits
16 bits
16 bits
8 bits
8 bits
variable
Frame Control
Destination Address
Source Address
Radius
Sequence Number
Frame Payload
Zigbee Application Layer Frame (APS)
8 bits
0 or 8 bits
0 or 8 bits
0 or 16 bits
0 or 8 bits
variable
Frame Control
Destination End Point
Cluster ID
Profile ID
Source End Point
Frame Payload


Zigbee Network Layer Frame (NWK)
16 bits
16 bits
16 bits
8 bits
8 bits
variable
Frame Control
Destination Address
Source Address
Radius
Sequence Number
Frame Payload
Routing Fields
Zigbee Frame Control
2 bits
3 bits
2 bits
1 bit
1 bit
7 bits
Frame Type
Protocol Version
Discover Route
Reserved
Security
Reserved
Zigbee Route Request Command Frame
8 bits
8 bits
8 bits
16 bits
8 bits
Command Frame Identifier
Command Options
Route Request Identifier
Destination Address
Path Cost
Zigbee Route Reply Command Frame
8 bits
8 bits
8 bits
16 bits
16 bits
8 bits
Command Frame Identifier
Command Options
Route Request Identifier
Originator Address
Responder Address
Path Cost

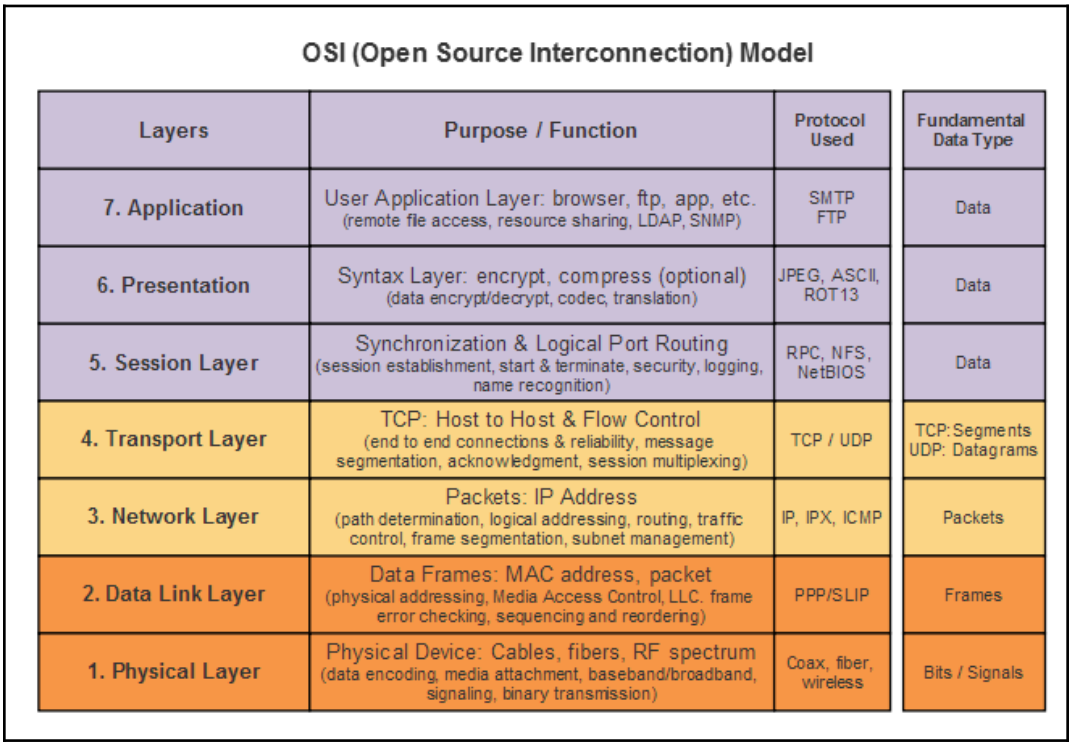

Z-Wave Protocol Stack
Simplified OSI Model
Application Layer
Routing Layer
(Routing and topology scans)
Transfer Layer
(Packet retransmission, ACK, checksums)
MAC Layer
(ITU-T G.9959)
(CSMA/CA, HomeID and NodeID Management)
PHY Layer
(ITU-T G.9959)
(908MHz / 860Mhz Radios)
7. Application Layer
6. Presentation Layer
5. Session Layer
4. Transport Layer
3. Network Layer
2. Data Link Layer
1. Physical Layer


ITU-T G.9959 PHY Frame
Variable - Max 64 bytes
Preamble
Start of Frame (SoF)
MAC Data Frame
End of Frame (EoF)
ITU-T G.9959 MAC Frame (Singlecast)
32 bits
8 bits
8 bits
8 bits
32 bits
8 bits
Variable
8 bits
Home ID
Source ID
Singlecast Header
Data Length
Destination ID
Hop Count
Repeater #1
Repeater #2
Data Payload
Checksum
ITU-T G.9959 MAC Frame (Routed)
Multicast Header
ITU-T G.9959 MAC Frame (Multicast)
Routed Header
ID Mask
Dest #1
Dest #2
Z-Wave Application Frame
Header
Command Class
Command
Parameter 1
Parameter 2
Parameter n



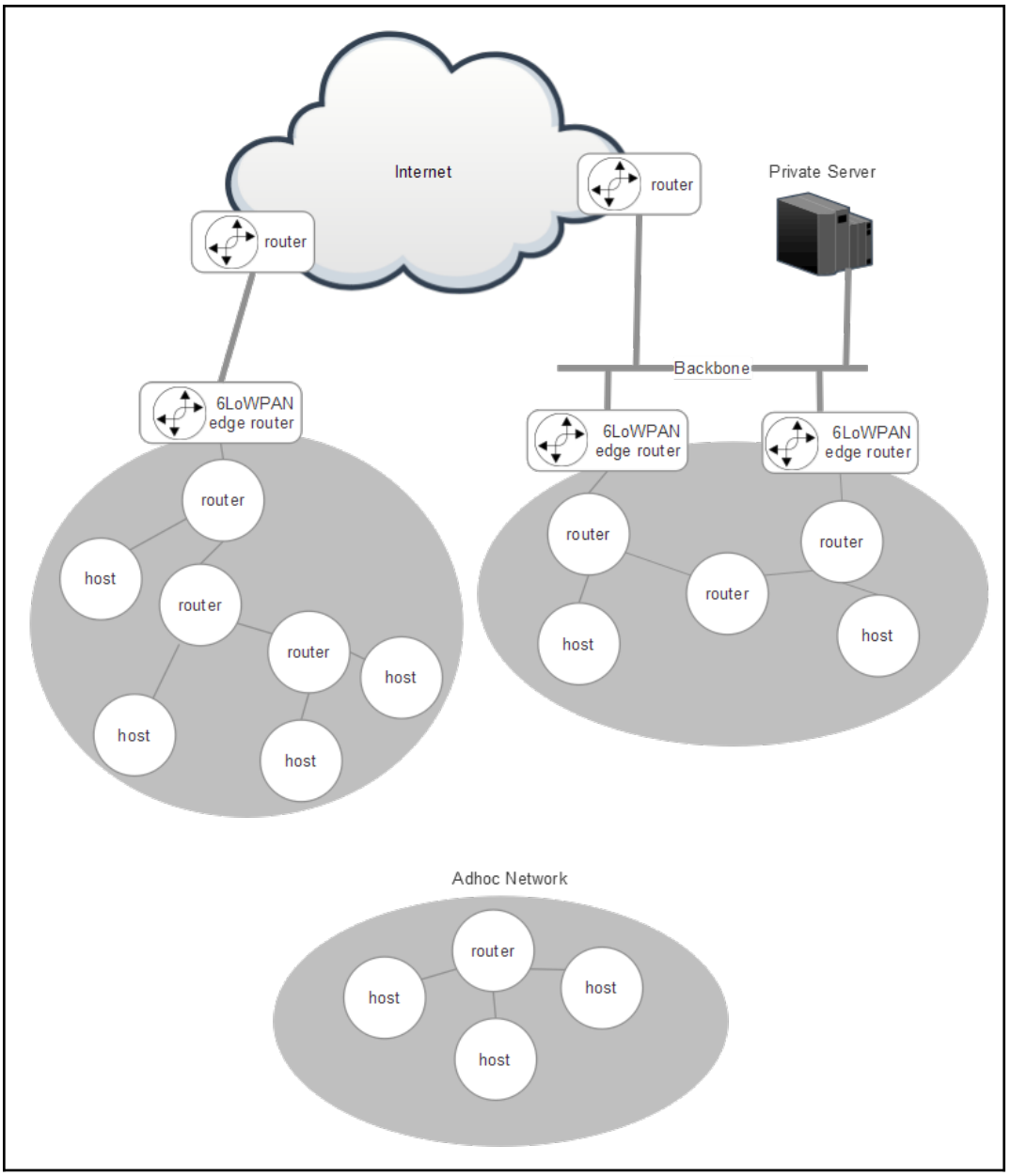
	Slave 1	Slave 2	Slave 3	Slave 4	Enhanced Slave 5	Primary Controller	Secondary SIS	Bridge	Portable Controller
Slave 1	0	1	1	0	0	1	0	0	0
Slave 2	1	0	1	0	1	0	0	0	1
Slave 3	1	1	0	0	0	0	0	1	0
Slave 4	0	0	0	0	1	0	0	0	0
Enhanced Slave 5	0	1	0	1	0	0	1	0	0
Primary Controller	0	0	0	0	0	0	0	0	0
Secondary SIS	0	0	0	0	1	0	0	0	0
Bridge	1	0	1	0	0	0	0	0	0
Portable Controller	0	1	0	0	0	0	0	0	0

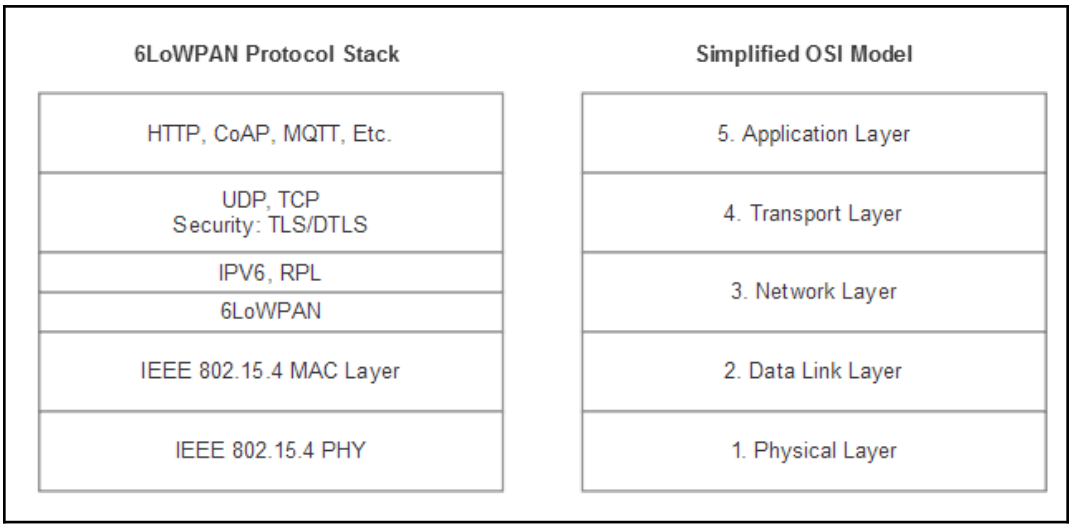
Chapter 6: IP-Based WPAN and WLAN

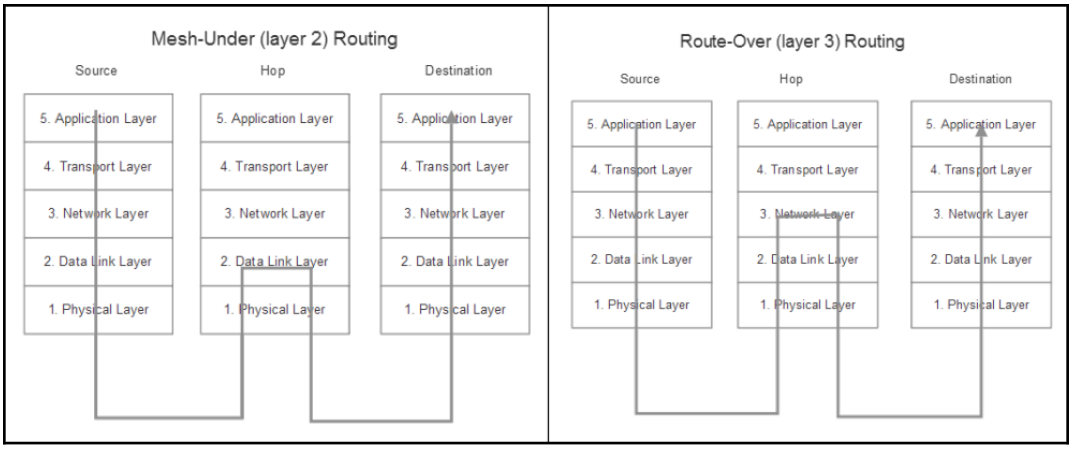


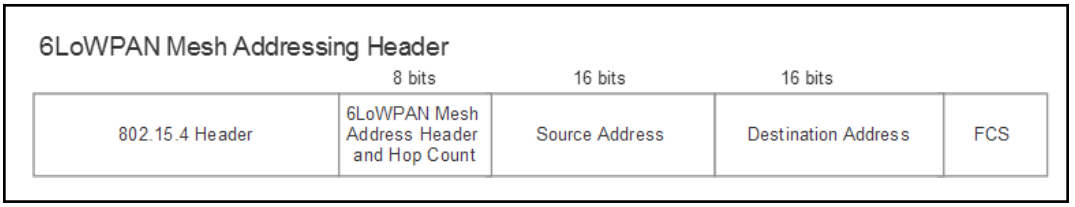
OSI (Open Source Interconnection) Model

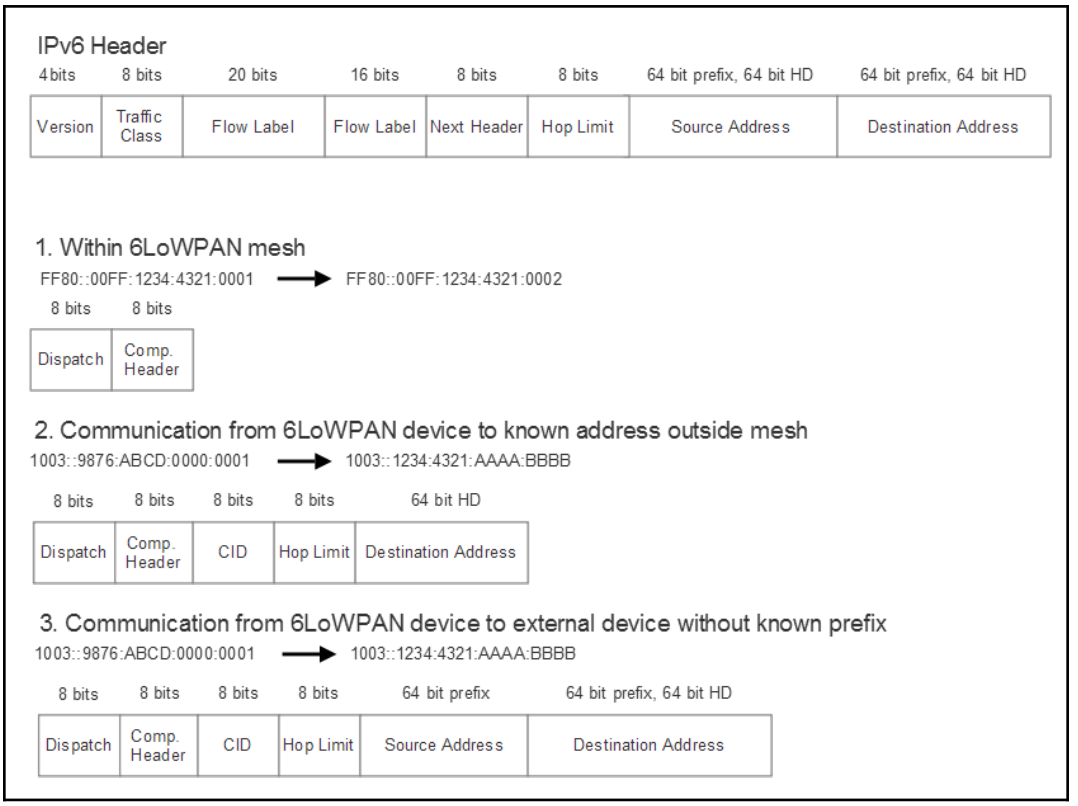
Layers	Purpose / Function	Protocol Used	Fundamental Data Type
7. Application	User Application Layer: browser, ftp, app, etc. (remote file access, resource sharing, LDAP, SNMP)	SMTP FTP	Data
6. Presentation	Syntax Layer: encrypt, compress (optional) (data encrypt/decrypt, codec, translation)	JPEG, ASCII, ROT13	Data
5. Session Layer	Synchronization & Logical Port Routing (session establishment, start & terminate, security, logging, name recognition)	RPC, NFS, NetBIOS	Data
4. Transport Layer	TCP: Host to Host & Flow Control (end to end connections & reliability, message segmentation, acknowledgment, session multiplexing)	TCP / UDP	TCP: Segments UDP: Datagrams
3. Network Layer	Packets: IP Address (path determination, logical addressing, routing, traffic control, frame segmentation, subnet management)	IP, IPX, ICMP	Packets
2. Data Link Layer	Data Frames: MAC address, packet (physical addressing, Media Access Control, LLC. frame error checking, sequencing and reordering)	PPP/SLIP	Frames
1. Physical Layer	Physical Device: Cables, fibers, RF spectrum (data encoding, media attachment, baseband/broadband, signaling, binary transmission)	Coax, fiber, wireless	Bits / Signals

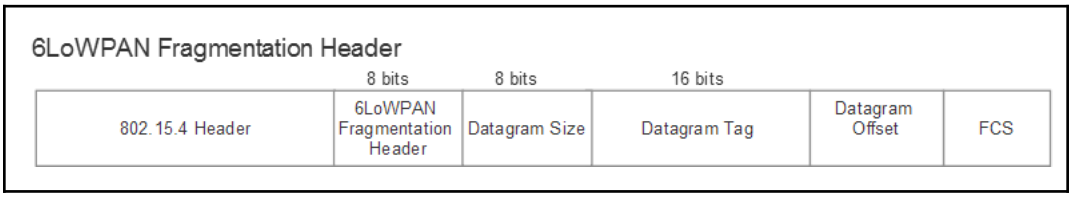

Internet
router
router
Private Server
Backbone
6LoWPAN
edge router
6LoWPAN
edge router
6LoWPAN
edge router
router
host
router
router
host
host
host
router
router
router
host
host
Adhoc Network
router
host
host
host

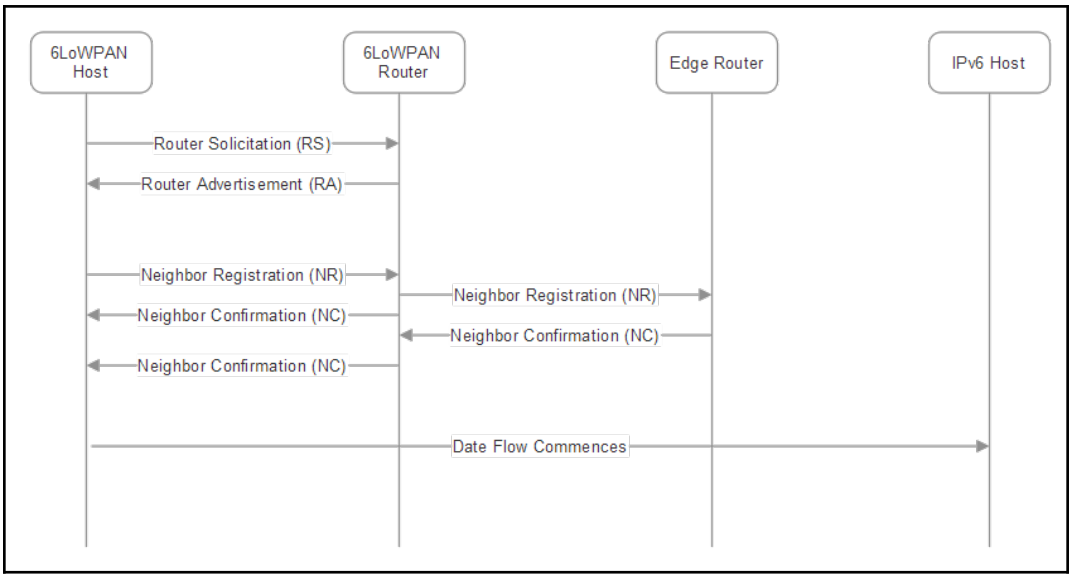

6LoWPAN Protocol Stack
HTTP, CoAP, MQTT, Etc.
UDP, TCP
Security: TLS/DTLS
IPV6, RPL
6LoWPAN
IEEE 802.15.4 MAC Layer
IEEE 802.15.4 PHY
Simplified OSI Model
5. Application Layer
4. Transport Layer
3. Network Layer
2. Data Link Layer
1. Physical Layer

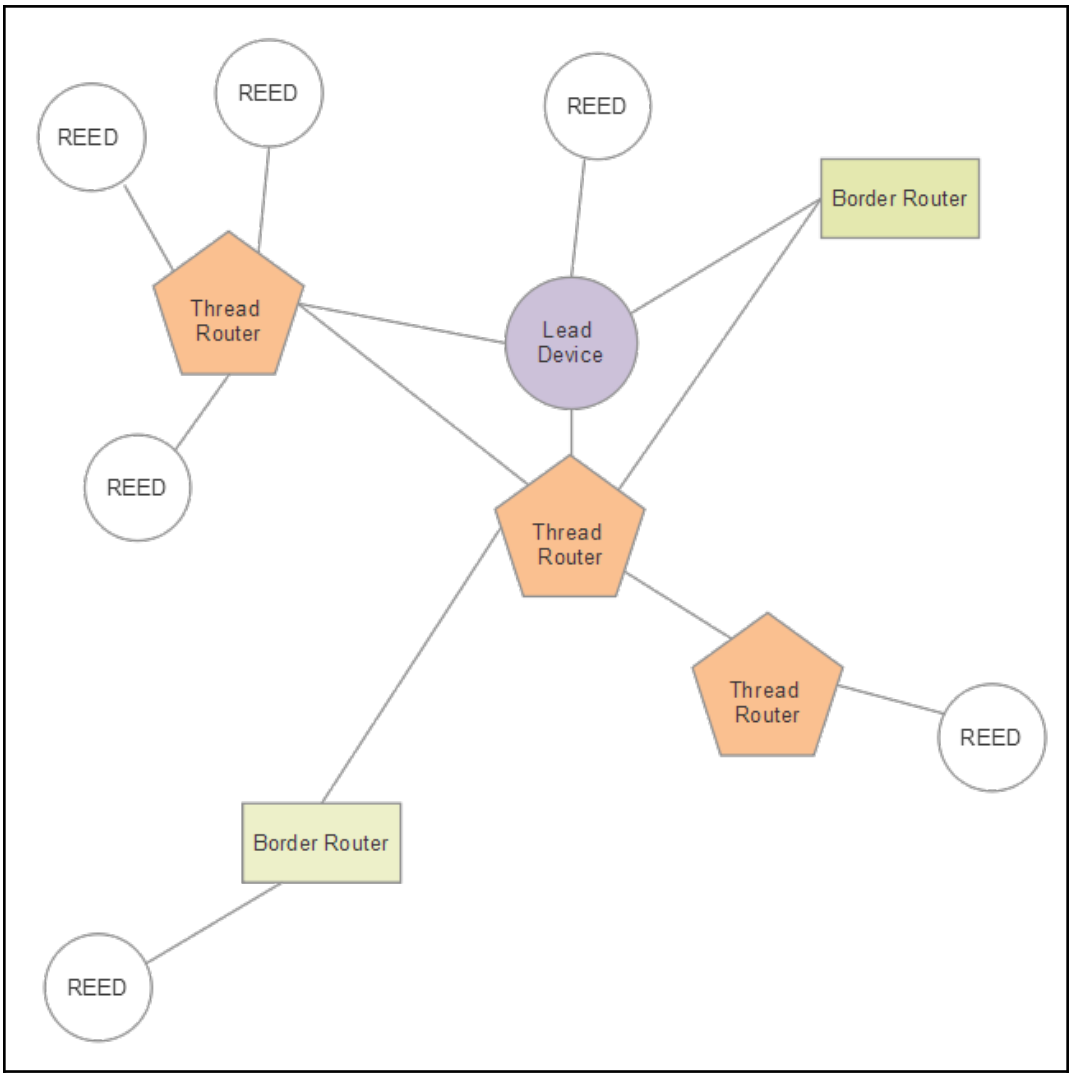

Mesh-Under (layer 2) Routing
Source
Hop
Destination
5. Application Layer
4. Transport Layer
3. Network Layer
2. Data Link Layer
1. Physical Layer
Route-Over (layer 3) Routing
Source
Hop
Destination
5. Application Layer
4. Transport Layer
3. Network Layer
2. Data Link Layer
1. Physical Layer

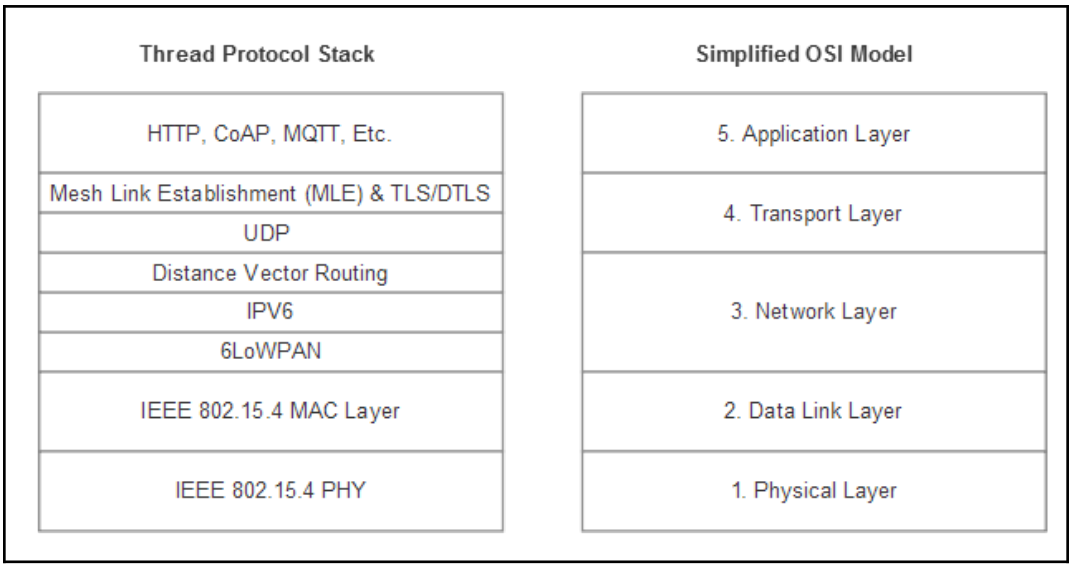

6LoWPAN Mesh Addressing Header
8 bits
16 bits
16 bits
802.15.4 Header
6LoWPAN Mesh Address Header and Hop Count
Source Address
Destination Address
FCS

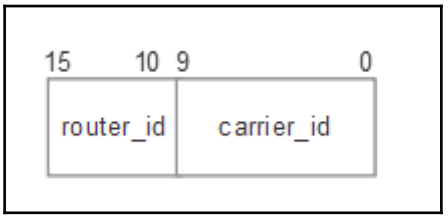

IPv6 Header
4 bits
8 bits
20 bits
16 bits
8 bits
8 bits
64 bit prefix, 64 bit HD
64 bit prefix, 64 bit HD
Version
Traffic Class
Flow Label
Flow Label
Next Header
Hop Limit
Source Address
Destination Address
1. Within 6LoWPAN mesh
FF80::00FF:1234:4321:0001
FF80::00FF:1234:4321:0002
8 bits
8 bits
Dispatch
Comp. Header
2. Communication from 6LoWPAN device to known address outside mesh
1003::9876:ABCD:0000:0001
1003::1234:4321:AAAA:BBBB
8 bits
8 bits
8 bits
8 bits
64 bit HD
Dispatch
Comp. Header
CID
Hop Limit
Destination Address
3. Communication from 6LoWPAN device to external device without known prefix
1003::9876:ABCD:0000:0001
1003::1234:4321:AAAA:BBBB
8 bits
8 bits
8 bits
8 bits
64 bit prefix
64 bit prefix, 64 bit HD
Dispatch
Comp. Header
CID
Hop Limit
Source Address
Destination Address


6LoWPAN Fragmentation Header
8 bits
8 bits
16 bits
802.15.4 Header
6LoWPAN Fragmentation Header
Datagram Size
Datagram Tag
Datagram Offset
FCS

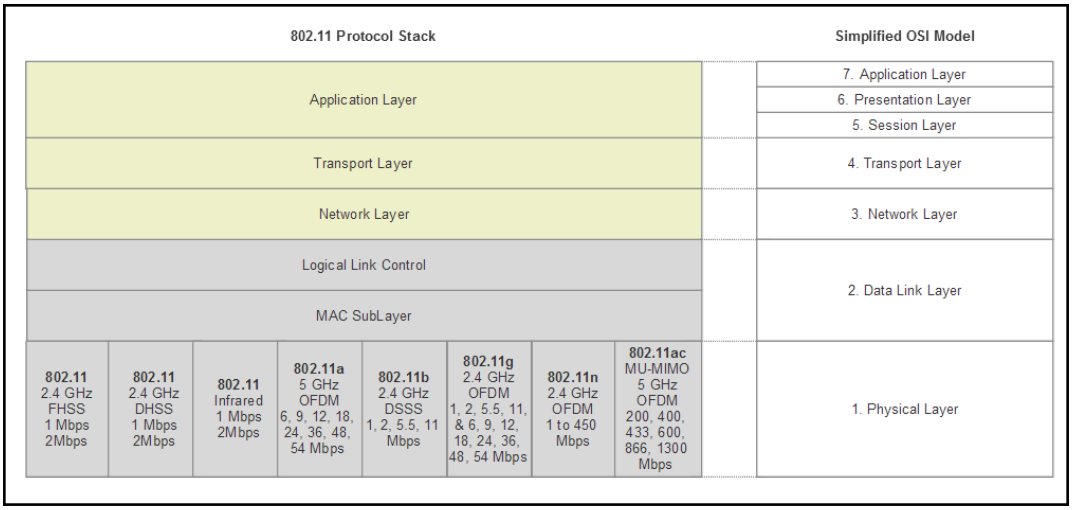

6LoWPAN Host
6LoWPAN Router
Edge Router
IPv6 Host
Router Solicitation (RS)
Router Advertisement (RA)
Neighbor Registration (NR)
Neighbor Registration (NR)
Neighbor Confirmation (NC)
Neighbor Confirmation (NC)
Neighbor Confirmation (NC)
Date Flow Commences

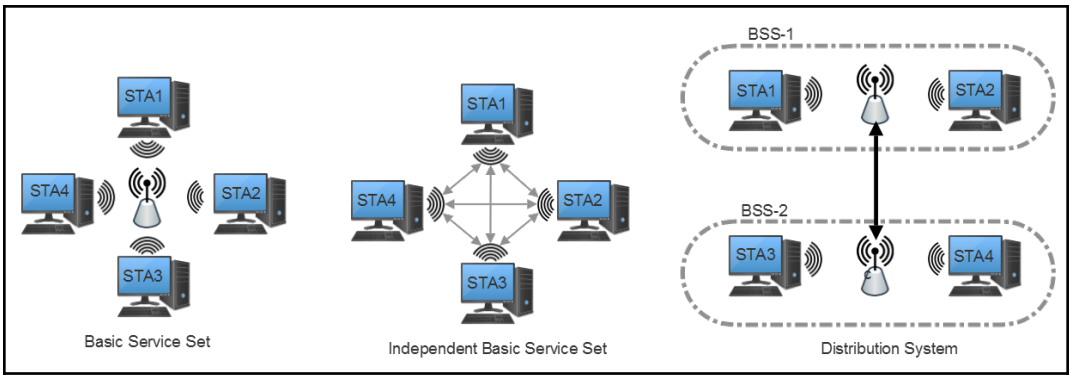

REED
REED
REED
REED
Border Router
Thread Router
Lead Device
REED
Thread Router
Thread Router
REED
Border Router
REED

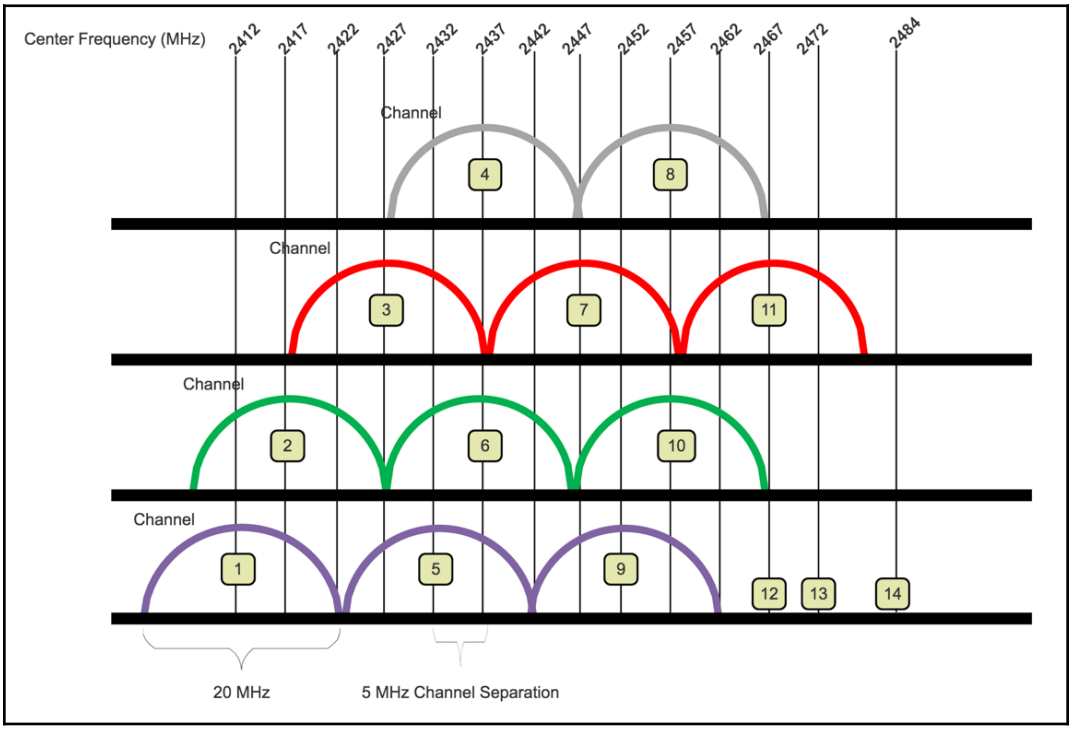

Thread Protocol Stack
HTTP, CoAP, MQTT, Etc.
Mesh Link Establishment (MLE) & TLS/DTLS
UDP
Distance Vector Routing
IPV6
6LoWPAN
IEEE 802.15.4 MAC Layer
IEEE 802.15.4 PHY
Simplified OSI Model
5. Application Layer
4. Transport Layer
3. Network Layer
2. Data Link Layer
1. Physical Layer

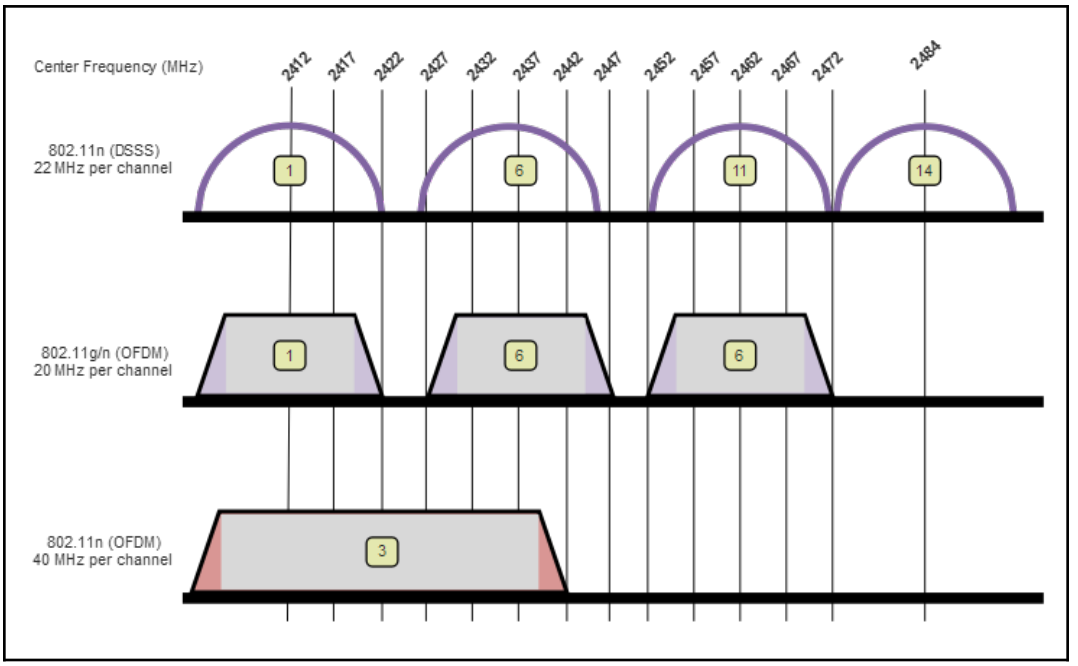

15
10
9
0
router_id
carrier_id

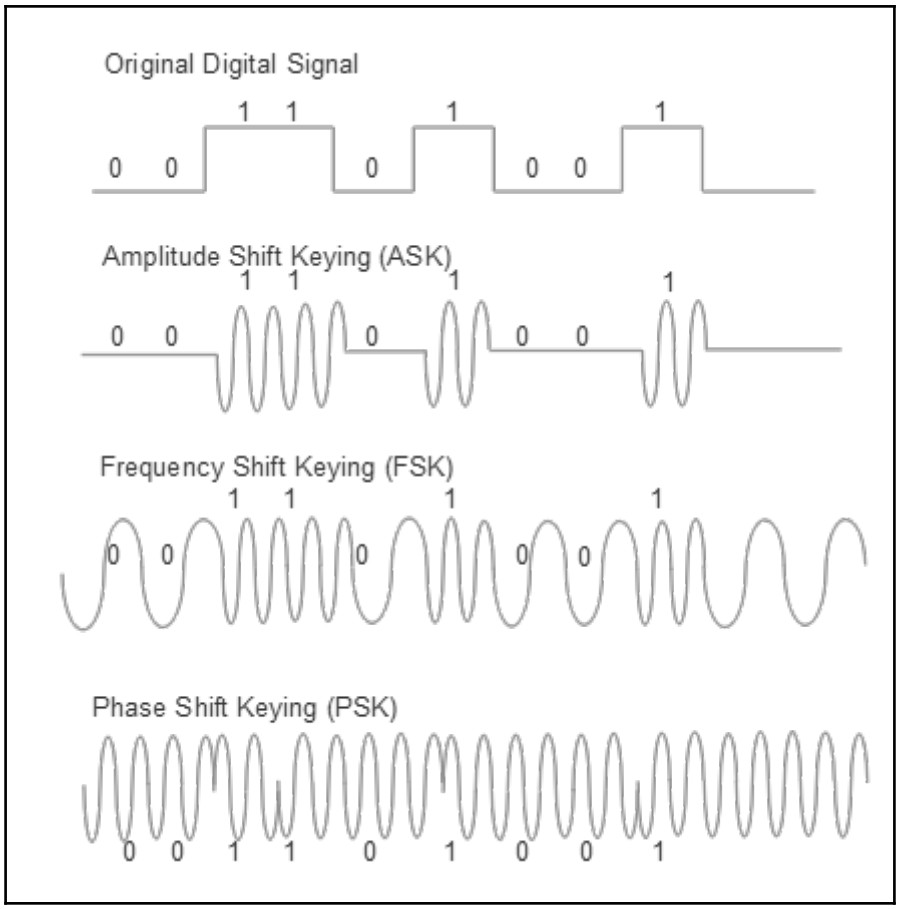
IEEE 802.11 Protocol	Use Case	Release Date	Frequency (GHz)	Bandwidth (MHz)	Streaming Data Rate per Channel min-max (Mbps)	Allowable MIMO Streams	Modulation	Indoor Range (m)	Outdoor Range (m)	Typical Dissipated Power per Chip (mW)
802.11	First 802.11 design	Jun-97	2.4	22	1 to 2	1	DSSS, FHSS	20	20	50
a	Release simulataneously with 802.11b Less prone to interefernce than 802.11b	Sep-99	5 3.7	20	6 to 54	1	OFDM (SISO)	30	120 5000	50
b	Release simulataneously with 802.11a Significant speed increase over 802.11a at improved range	Sep-99	2.4	22	1 to 11	1	DSSS (SISO)	50	150	7 to 50
g	Speed increase over 802.11b	Jun-03	2.4	20	6 to 54	1	OFDM. DSSS (SISO)	38	140	50
n	Multiple antenna technology for improved speed, and range.	Oct-09	2.4 / 5	20 40	7.2 to 72.2 15 to 150	4	OFDM (MIMO)	70	250	40
ac	Better performacne and coverage over 802.11n. Wider channel and improved modulation. Allows multiple users using MU-MIMO. Introduced beamforming.	Dec-13	5	20 40 80 160	7.2 to 96.3 15 to 200 32.5 to 433.3 65 to 866.7	8	OFDM (MU_MIMO)	35	35	40
ah	"WiFi HaLow" Designed for IoT and sensor networks. Very low power and wider range.	Dec-16	2.4 / 5	1 to 16	347	4	OFDM	1000	1000	tbd but goal is low power
p	"Wireless Access in Vehicular Environments" "Intelligent Transport Systems" Dedicated Short Range Communication Trnasport uses cases: toll collection, safety and collision emergencies, vehicular networking.	Jun-09	5.9	10	27	1	OFDM	NA	400 to 1000	40
af	"white WiFi" or "Super WiFi" Deploy unused spectrum in TV bands to provide last mile connectivity in India, Kenya, Singapoere, US and UK	Nov-13	0.470 to 0.710	6 to 8	568	4	OFDM	NA	6000-100,000	tbd
ad	WiGig Alliance 60 GHz Wireless for HD video and projectors Audio and video transport and cable replacement	Dec-12	60	2160	4260	>10	SC, OFDM (MU-MIMO)	10	10	tbd
ax	"High Efficiency Wireless (HEW)" Next gen 802.11 4x increase in capacity over 802.11ac Average increase of 4x speed per user over 802.11ac Backwards compatible to 802.11a/b/g/n/ac Dense deployment scenarios	2019	2.4 / 5	20 40 80 160	450 to 10000	8	OFDMA (MU-MIMO)	35	35	tbd

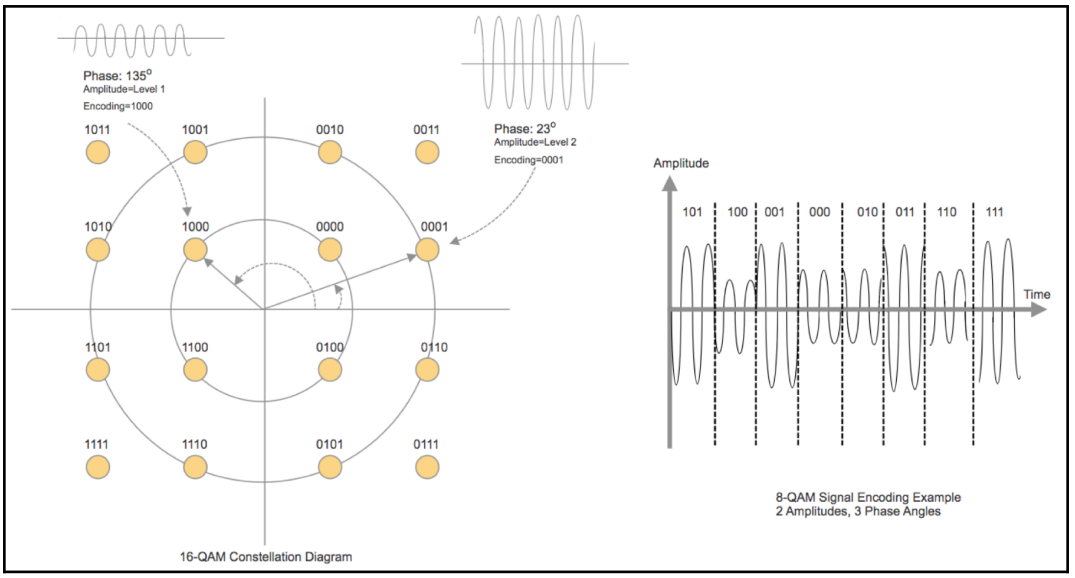


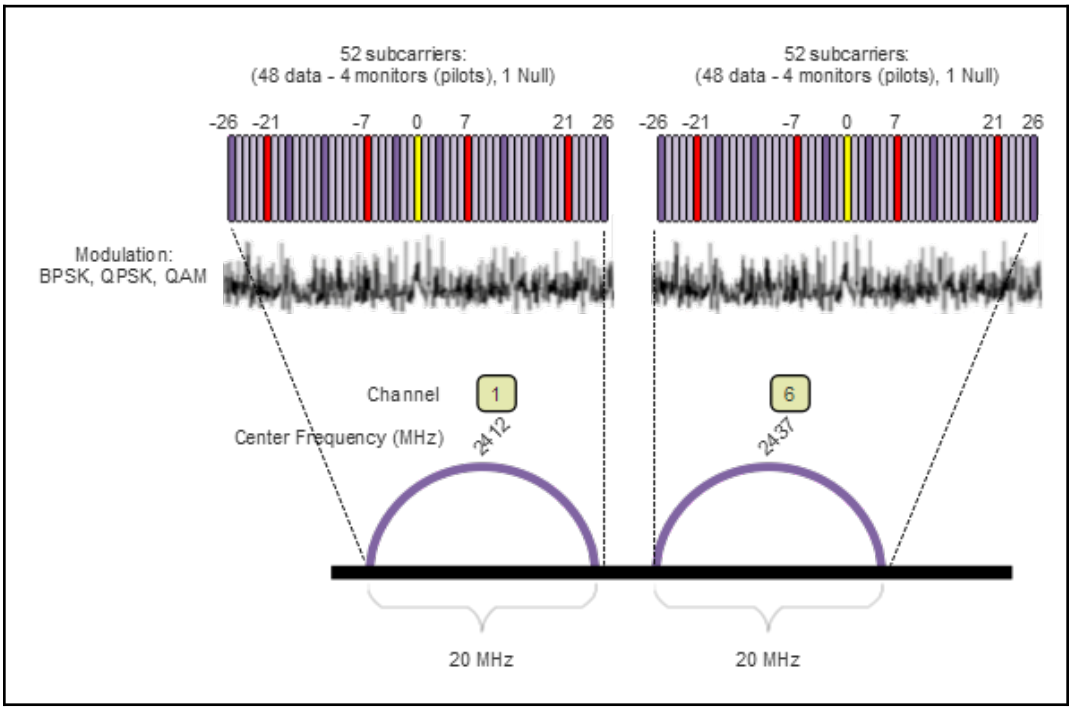

STA1
STA4
STA2
STA3
Basic Service Set
STA1
STA4
STA2
STA3
Independent Basic Service Set
BSS-1
STA1
STA2
BSS-2
STA3
STA4
Distribution System

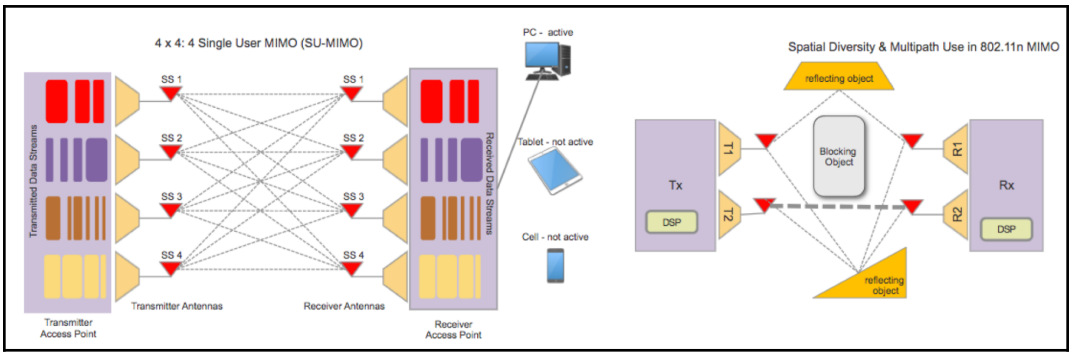

Center Frequency (MHz)
2412
2417
2422
2427
2432
2437
2442
2447
2452
2457
2462
2467
2472
2484
Channel
4
8
Channel
3
7
11
Channel
2
6
10
Channel
1
5
9
12
13
14
20 MHz
5 MHz Channel Separation

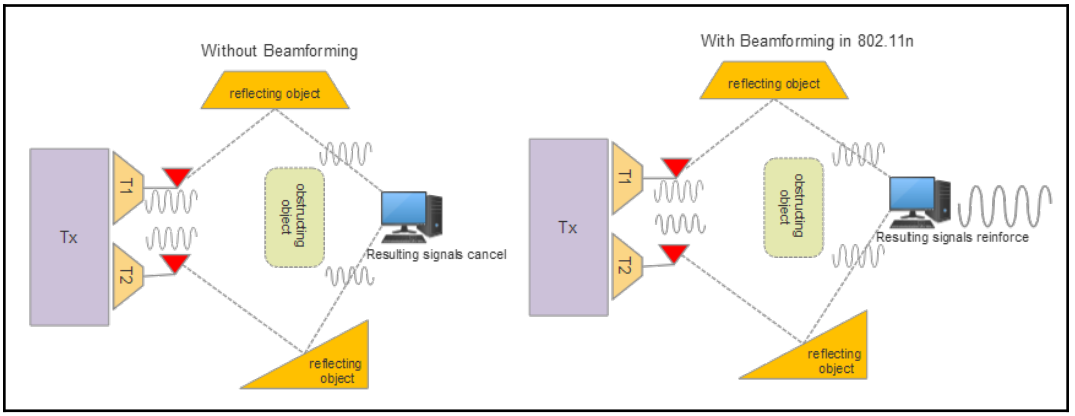

Center Frequency (MHz)
2412
2417
2422
2427
2432
2437
2442
2447
2452
2457
2462
2467
2472
2484
802.11n (DSSS)
22 MHz per channel
1
6
11
14
802.11g/n (OFDM)
20 MHz per channel
1
6
6
802.11n (OFDM)
40 MHz per channel
3

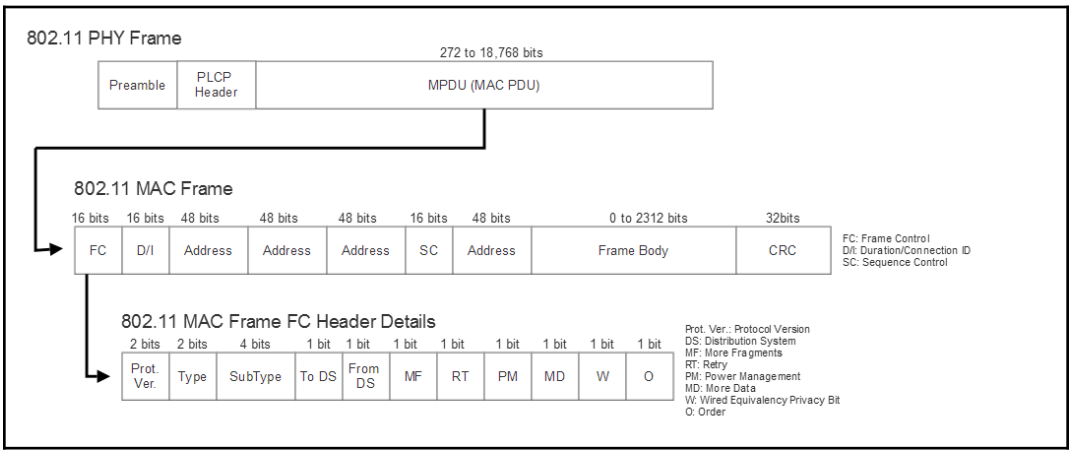

Original Digital Signal
1 1 1 1
0 0 0 0 0
Amplitude Shift Keying (ASK)
1 1 1 1
0 0 0 0 0
Frequency Shift Keying (FSK)
1 1 1 1
0 0 0 0 0
Phase Shift Keying (PSK)
0 0 1 1 0 1 0 0 1

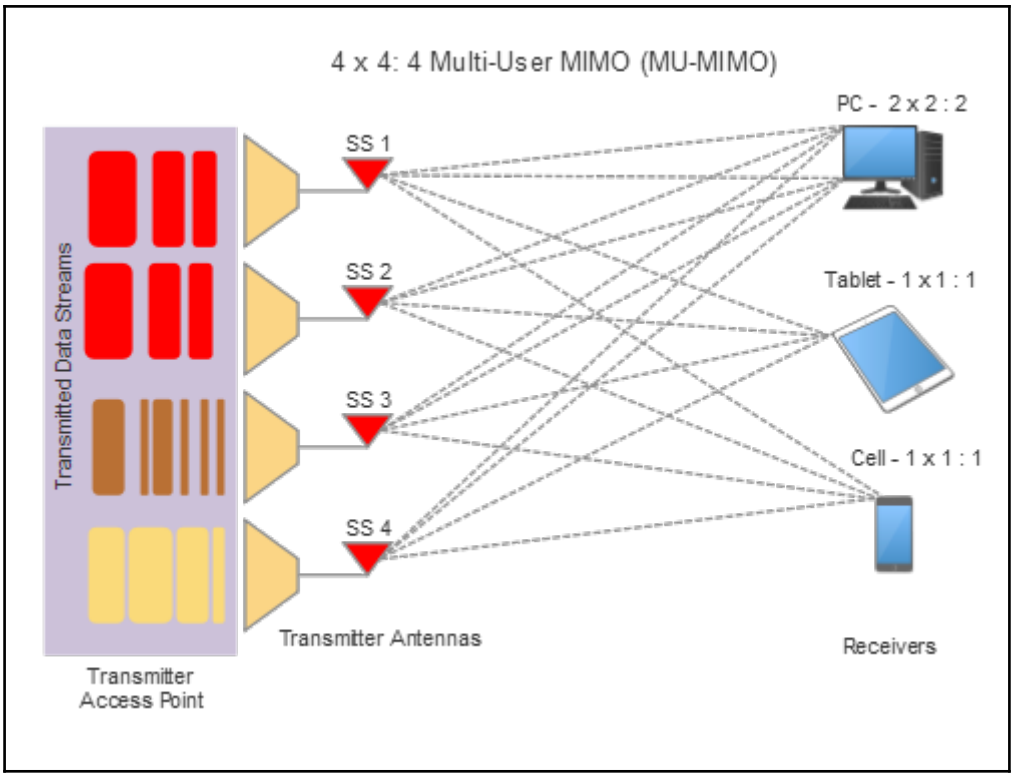

Phase: 135°
Amplitude=Level 1
Encoding=1000
1011
1001
0010
0011
Phase: 23°
Amplitude=Level 2
Encoding=0001
1010
1000
0000
0001
1101
1100
0100
0110
1111
1110
0101
0111
16-QAM Constellation Diagram
Amplitude
101
100
001
000
010
011
110
111
Time
8-QAM Signal Encoding Example
2 Amplitudes, 3 Phase Angles

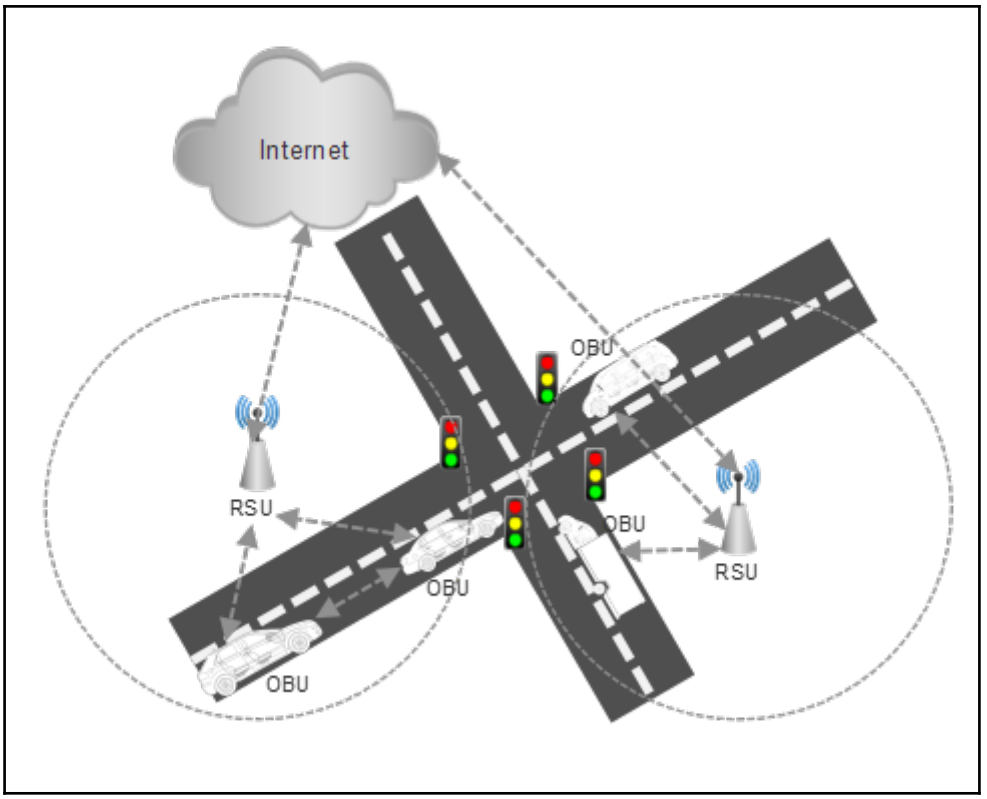

52 subcarriers:
(48 data - 4 monitors (pilots), 1 Null)
52 subcarriers:
(48 data - 4 monitors (pilots), 1 Null)
-26 -21 -7 0 7 21 26
-26 -21 -7 0 7 21 26
Modulation:
BPSK, QPSK, QAM
Channel
1
6
Center Frequency (MHz)
2412
2437
20 MHz
20 MHz


4 x 4: 4 Single User MIMO (SU-MIMO)
Transmitted Data Streams
SS 1
SS 2
SS 3
SS 4
Transmitter Antennas
Receiver Antennas
Received Data Streams
Transmitter Access Point
Receiver Access Point
PC - active
Tablet - not active
Cell - not active
Spatial Diversity & Multipath Use in 802.11n MIMO
reflecting object
Tx
DSP
T1
T2
Blocking Object
R1
R2
Rx
DSP
reflecting object

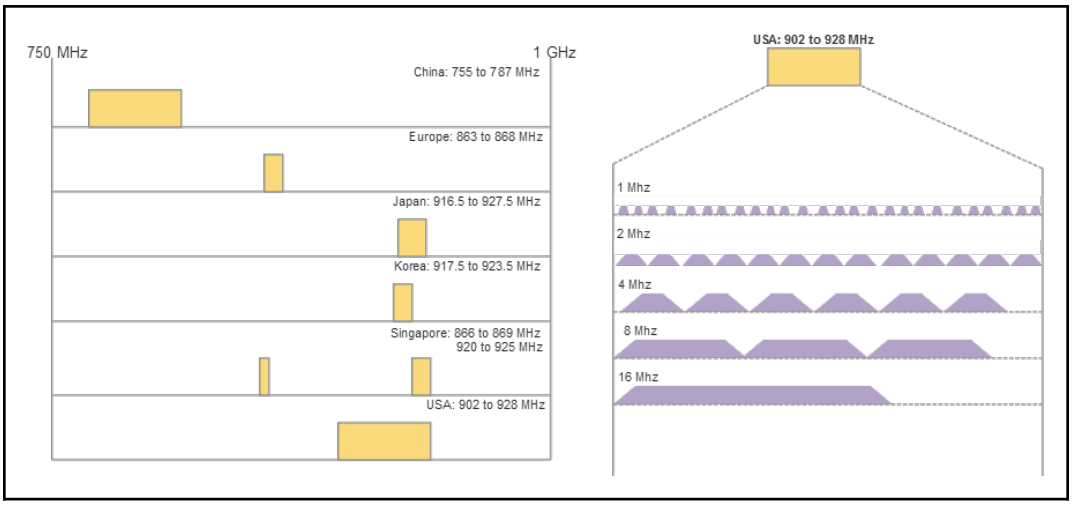

Without Beamforming
reflecting object
Tx
T1
T2
obstructing object
Resulting signals cancel
reflecting object
With Beamforming in 802.11n
reflecting object
Tx
T1
T2
obstructing object
Resulting signals reinforce
reflecting object

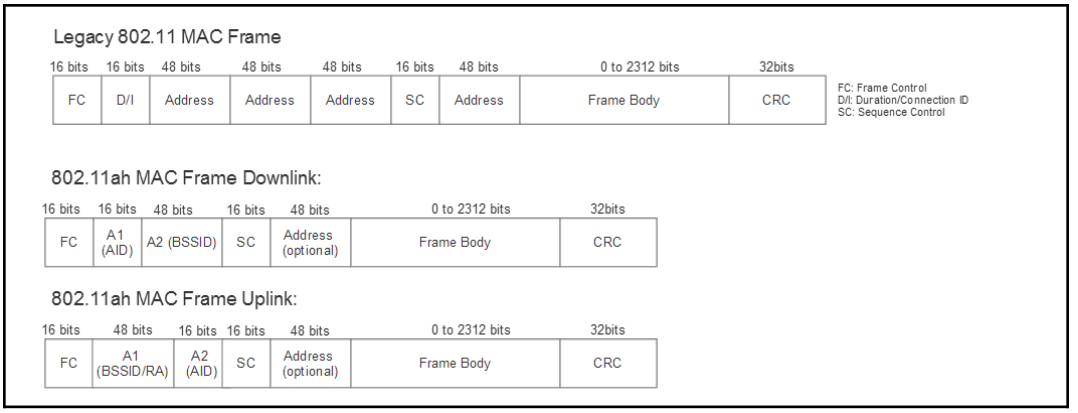

802.11 PHY Frame
272 to 18,768 bits
Preamble
PLCP Header
MPDU (MAC PDU)
802.11 MAC Frame
16 bits
16 bits
48 bits
48 bits
48 bits
16 bits
48 bits
0 to 2312 bits
32bits
FC
D/I
Address
Address
Address
SC
Address
Frame Body
CRC
FC: Frame Control
D/I: Duration/Connection ID
SC: Sequence Control
802.11 MAC Frame FC Header Details
2 bits
2 bits
4 bits
1 bit
1 bit
1 bit
1 bit
1 bit
1 bit
1 bit
1 bit
Prot. Ver.
Type
SubType
To DS
From DS
MF
RT
PM
MD
W
O
Prot. Ver.: Protocol Version
DS: Distribution System
MF: More Fragments
RT: Retry
PM: Power Management
MD: More Data
W: Wired Equivalency Privacy Bit
O: Order

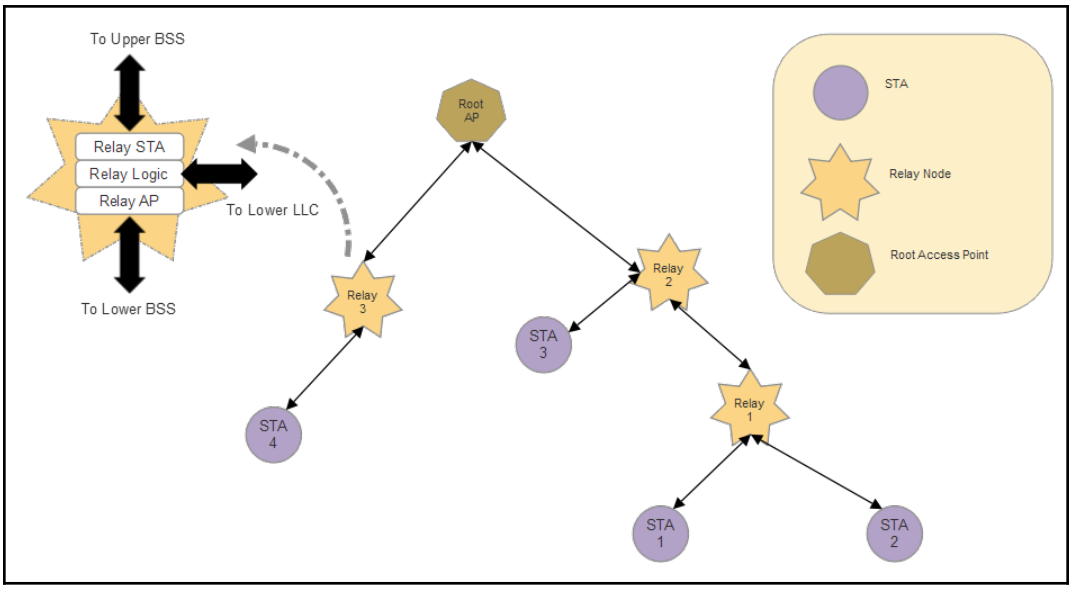

4 x 4: 4 Multi-User MIMO (MU-MIMO)
PC - 2 x 2 : 2
SS 1
SS 2
Tablet - 1 x 1 : 1
SS 3
Cell - 1 x 1 : 1
SS 4
Transmitted Data Streams
Transmitter Antennas
Receivers
Transmitter
Access Point

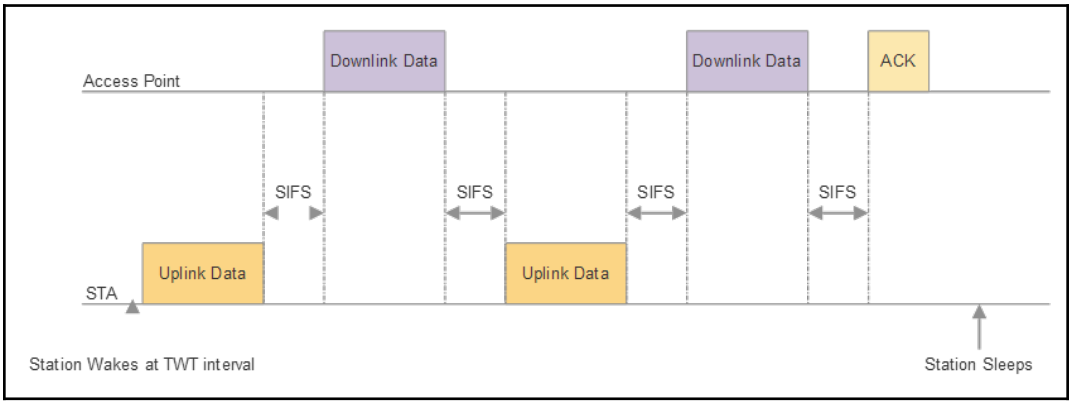


802.11 Protocol Stack		Simplified OSI Model
IEEE 1609.1 (Safety and Traffic Efficiency Applications)		7. Application Layer
IEEE 1609.2 WAVE Security Services		4. Transport Layer
TCP/UDP	IEEE 1609.3 WSMP	
IPV6		3. Network Layer
Logical Link Control		2. Data Link Layer
IEEE 1609.4 MAC SubLayer		
802.11p 5 GHz OFDM 3, 4.5, 6, 9, 12, 18, 24, 27 Mbps		1. Physical Layer

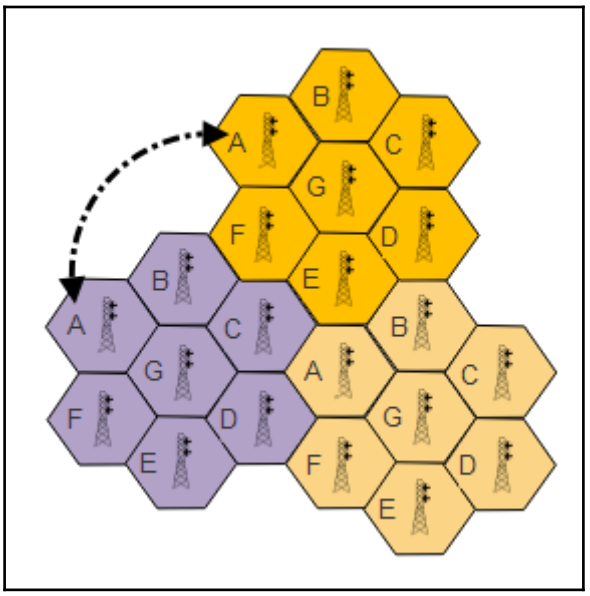

750 MHz
1 GHz
China: 755 to 787 MHz
Europe: 863 to 868 MHz
Japan: 916.5 to 927.5 MHz
Korea: 917.5 to 923.5 MHz
Singapore: 866 to 869 MHz
920 to 925 MHz
USA: 902 to 928 MHz
USA: 902 to 928 MHz
1 Mhz
2 Mhz
4 Mhz
8 Mhz
16 Mhz

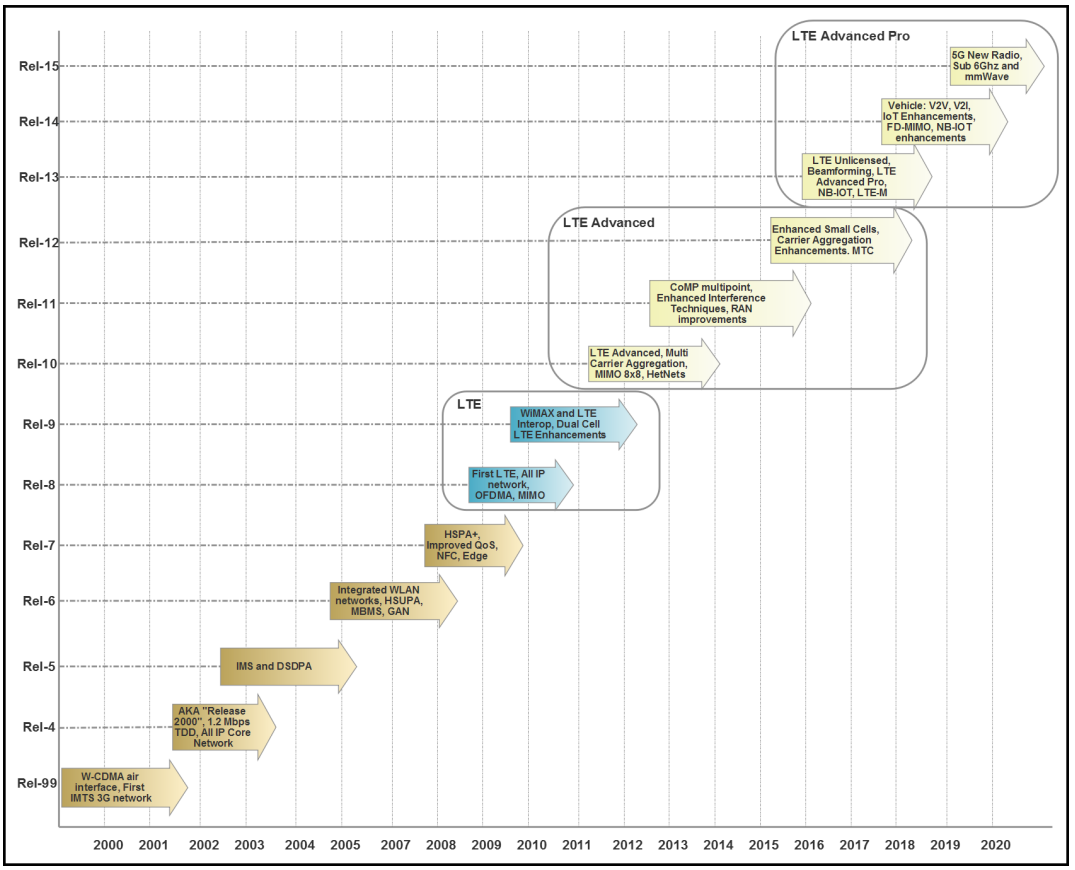

Legacy 802.11 MAC Frame
16 bits
16 bits
48 bits
48 bits
48 bits
16 bits
48 bits
0 to 2312 bits
32bits
FC
D/I
Address
Address
Address
SC
Address
Frame Body
CRC
FC: Frame Control
D/I: Duration/Connection ID
SC: Sequence Control
802.11ah MAC Frame Downlink:
16 bits
16 bits
48 bits
16 bits
48 bits
0 to 2312 bits
32bits
FC
A1 (AID)
A2 (BSSID)
SC
Address (optional)
Frame Body
CRC
802.11ah MAC Frame Uplink:
16 bits
48 bits
16 bits
16 bits
48 bits
0 to 2312 bits
32bits
FC
A1 (BSSID/RA)
A2 (AID)
SC
Address (optional)
Frame Body
CRC

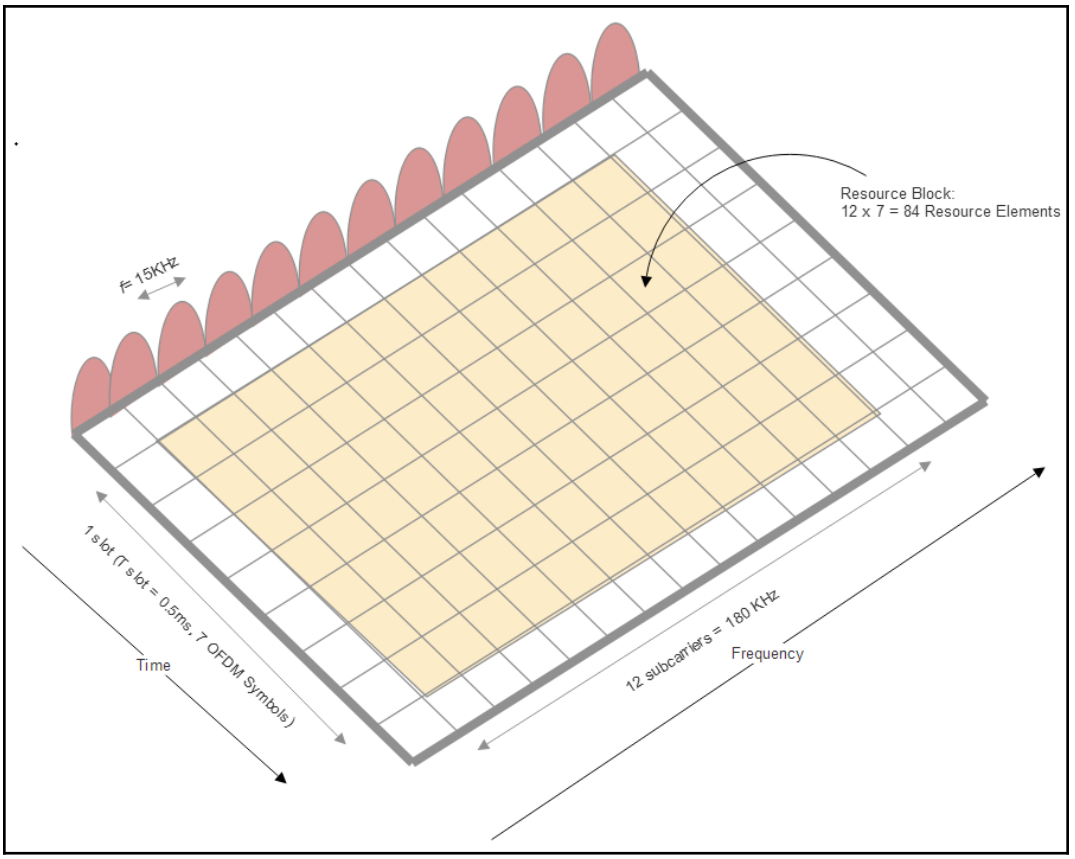

To Upper BSS
Relay STA
Relay Logic
Relay AP
To Lower LLC
To Lower BSS
Root AP
Relay 3
Relay 2
Relay 1
STA 3
STA 4
STA 1
STA 2
STA
Relay Node
Root Access Point


Access Point
Downlink Data
Downlink Data
ACK
SIFS
SIFS
SIFS
SIFS
STA
Uplink Data
Uplink Data
Station Wakes at TWT interval
Station Sleeps

Chapter 7: Long-Range Communication Systems and Protocols (WAN)

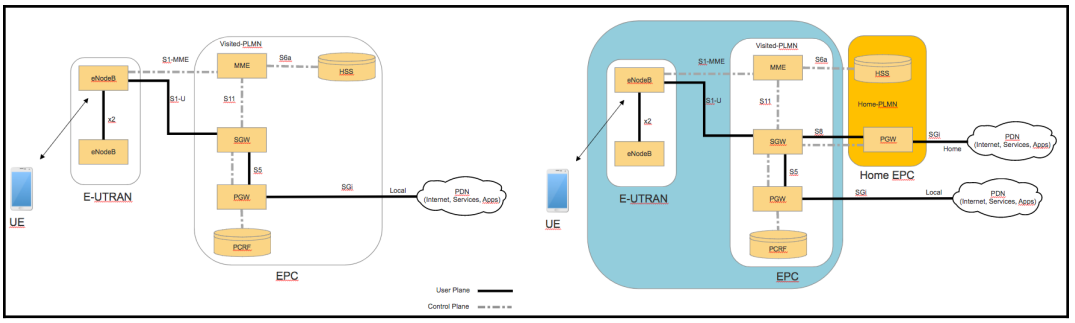


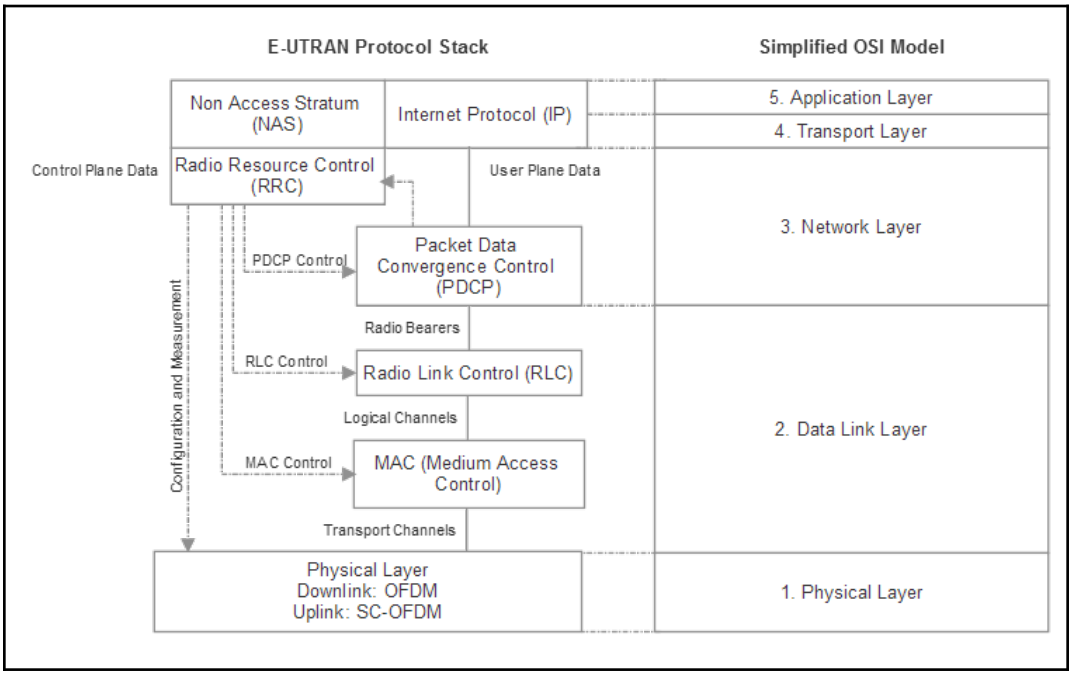

LTE Advanced Pro
LTE Advanced
LTE
Rel-15
Rel-14
Rel-13
Rel-12
Rel-11
Rel-10
Rel-9
Rel-8
Rel-7
Rel-6
Rel-5
Rel-4
Rel-99
5G New Radio, Sub 6Ghz and mmWave
Vehicle: V2V, V2I, IoT Enhancements, FD-MIMO, NB-IOT enhancements
LTE Unlicensed, Beamforming, LTE Advanced Pro, NB-IOT, LTE-M
Enhanced Small Cells, Carrier Aggregation Enhancements. MTC
CoMP multipoint, Enhanced Interference Techniques, RAN improvements
LTE Advanced, Multi Carrier Aggregation, MIMO 8x8, HetNets
WiMAX and LTE Interop, Dual Cell LTE Enhancements
First LTE, All IP network, OFDMA, MIMO
HSPA+, Improved QoS, NFC, Edge
Integrated WLAN networks, HSUPA, MBMS, GAN
IMS and DSDPA
AKA "Release 2000", 1.2 Mbps TDD, All IP Core Network
W-CDMA air interface, First IMTS 3G network
2000 2001 2002 2003 2004 2005 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020

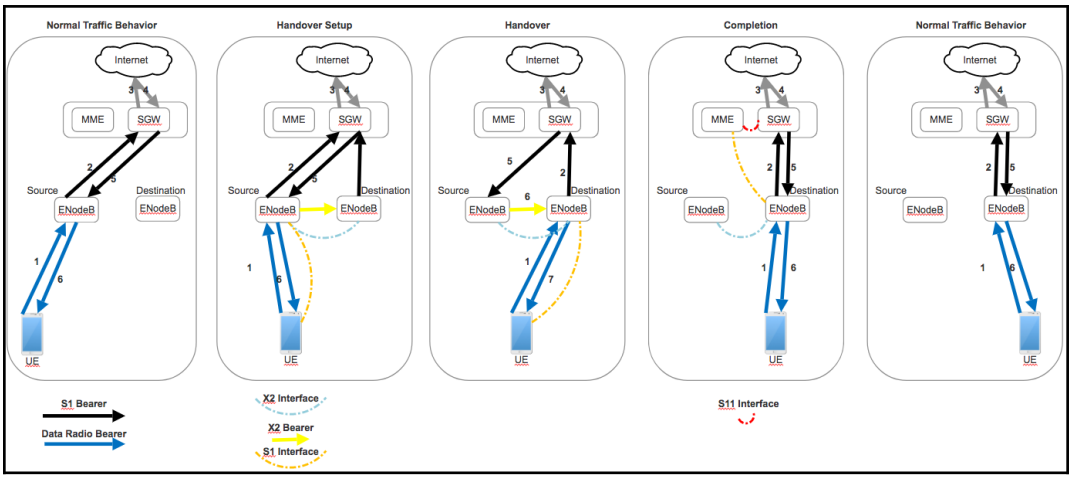

Resource Block:
12 x 7 = 84 Resource Elements
f= 15KHz
1 slot (T slot = 0.5ms, 7 OFDM Symbols)
Time
12 subcarriers = 180 KHz
Frequency

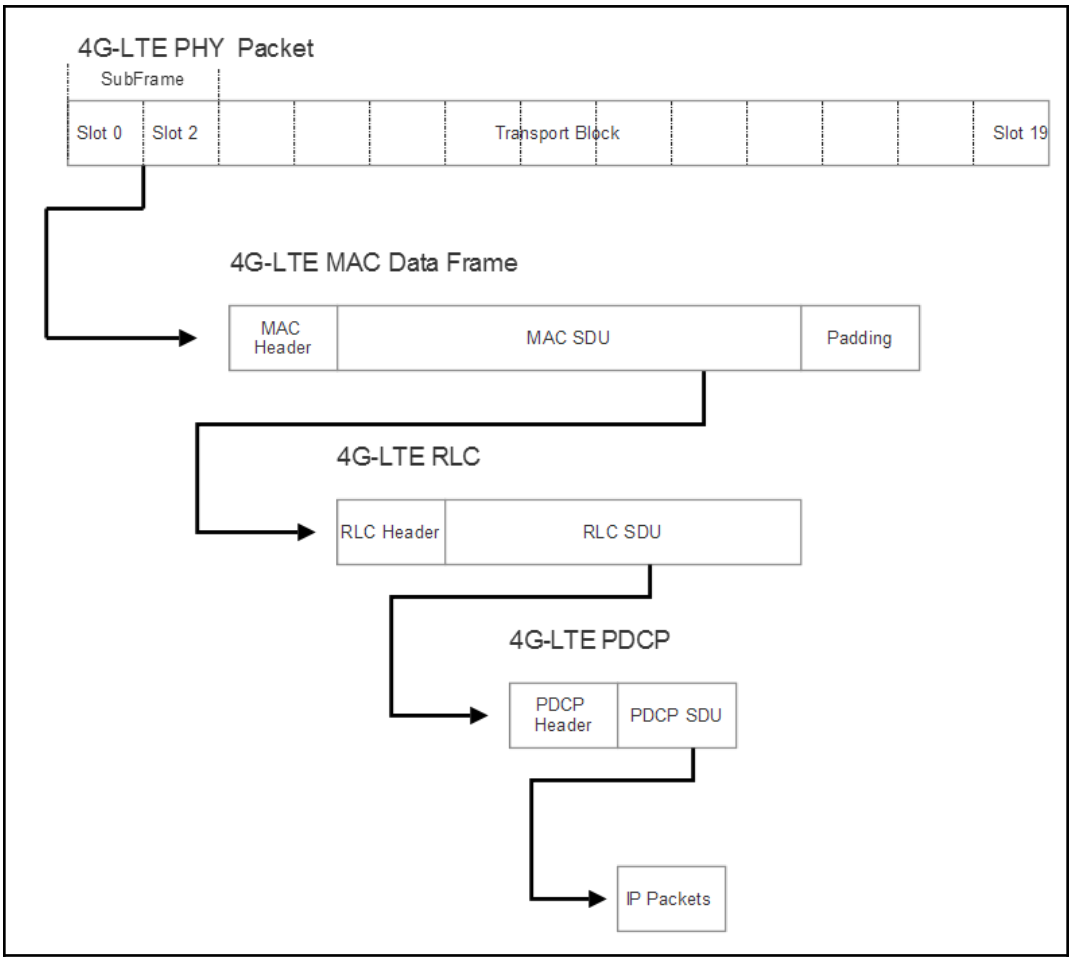
Band	Duplex	*f* (MHz)	Common Name	North America	Width of Band (MHz)	Duplex Space (MHz)	Band Gap (MHz)
1	FDD	2100	IMT		60	190	130
2	FDD	1900	PCS blocks A-F	Yes	60	80	20
3	FDD	1800	DCS		75	95	20
4	FDD	1700	AWS blocks A-F (AWS-1)	Yes	45	400	355
5	FDD	850	CLR	USA (AT&T, U.S. Cellular)	25	45	20
6	FDD				10	35	25
7	FDD	2600	IMT-E	Canada (Bell, Rogers, Telus)	70	120	50
8	FDD	900	E-GSM		35	45	10
9	FDD				35	95	60
10	FDD	1700	Extended AWS blocks A-I		60	400	340
11	FDD	1500	Lower PDC	Canada (Bell), Guam (iConnect, ...	20	48	28
12	FDD	700	Lower SMH blocks A/B/C	USA (Verizon), Canada (Bell, EastLink, ...	18	30	12
13	FDD	700	Upper SMH block C	USA (FirstNet)	10	–31	41
14	FDD	700	Upper SMH block D		10	–30	40
15	FDD	2000			20	700	680
16	FDD	700			15	575	560
17	FDD	700	Lower SMH blocks B/C	Canada (Rogers), Guam (NTT), USA (AT&T)	12	30	18
18	FDD	850	Japan lower 800		15	45	30
19	FDD	850	Japan upper 800		15	45	30
20	FDD	800	EU Digital Dividend		30	–41	71
21	FDD	1500	Upper PDC		15	48	33
22	FDD	3500		USA (Ligado Networks)	90	100	10
23	FDD	2000			20	180	160
24	FDD	1600	L-Band (US)		34	–101.5	135.5
25	FDD	1900	Extended PCS blocks A-G	USA (Sprint)	65	80	15
26	FDD	850	Extended CLR	USA (Sprint)	30 / 40		10
27	FDD	800	SMR		17	45	28
28	FDD	700	APT		45	55	10
29	FDD[A 1]	700	Lower SMH blocks D/E	USA (AT&T)	11	n/a	
30	FDD	2300	WCS blocks A/B	USA (AT&T)	10	45	35
31	FDD	450			5	10	5
32	FDD[A 1]	1500	L-Band (EU)		44	n/a	
65	FDD	2100	Extended IMT		90	190	
66	FDD	1700	Extended AWS blocks A-J	Canada (Freedom Mobile)	90/70	400	
67	FDD[A 1]	700	EU 700		20	n/a	
68	FDD	700	ME 700		30	55	
69	FDD[A 1]	2600	IMT-E (Duplex spacing)		50	n/a	
70	FDD	2000	AWS-4	USA (DISH)	25/15	300	
71	FDD	600	US Digital Dividend	USA (T-Mobile)			

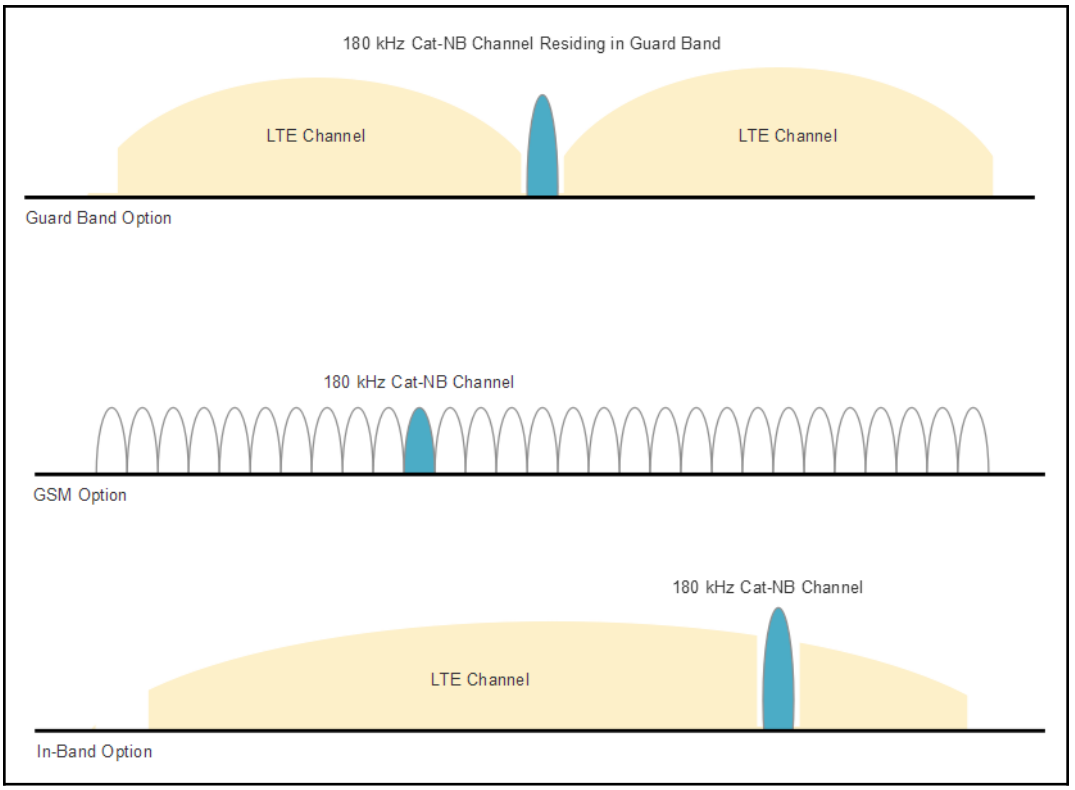
Band	Duplex	*f* (MHz)	Common Name	North America	Allocation (MHz)	Width of Band (MHz)
33	TDD	2100	IMT		1900 – 1920	20
34	TDD	2100	IMT		2010 – 2025	15
35	TDD	1900	PCS (Uplink)		1850 – 1910	60
36	TDD	1900	PCS (Downlink)		1930 – 1990	60
37	TDD	1900	PCS (Duplex spacing)		1910 – 1930	20
38	TDD	2600	IMT-E (Duplex Spacing)		2570 – 2620	50
39	TDD	1900	DCS-IMT gap		1880 – 1920	40
40	TDD	2300			2300 – 2400	100
41	TDD	2500	BRS / EBS	USA (Sprint)	2496 – 2690	194
42	TDD	3500			3400 – 3600	200
43	TDD	3700			3600 – 3800	200
44	TDD	700	APT		703 – 803	100
45	TDD	1500	L-Band (China)		1447 – 1467	20
46	TDD	5200	U-NII		5150 – 5925 (unlicensed)	775
47	TDD	5900	U-NII-4 (V2X)		5855 – 5925 (unlicensed)	70
48	TDD	3600	CBRS		3550 – 3700	150

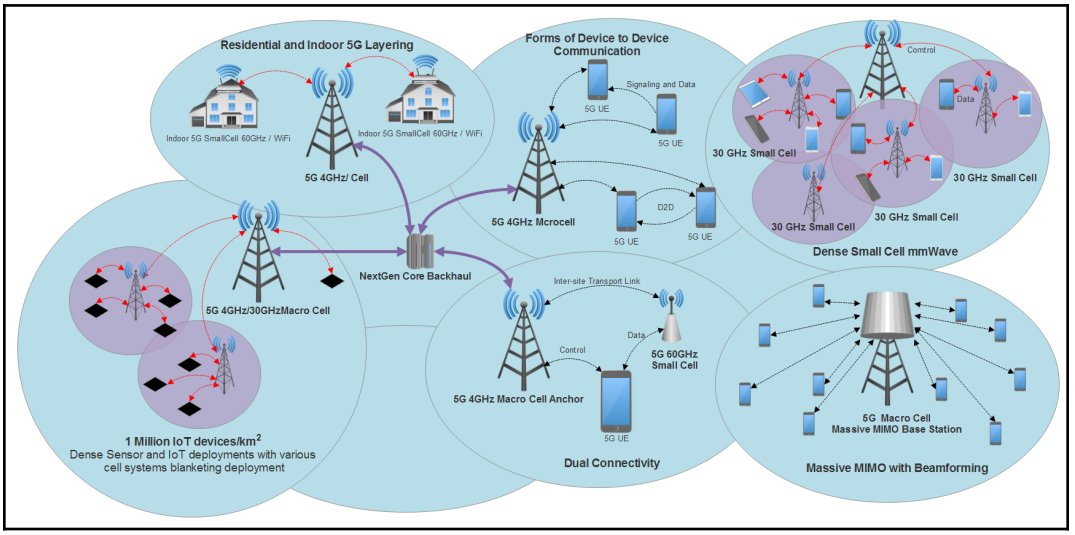

Visited-PLMN
S1-MME
S6a
MME
HSS
eNodeB
S1-U
S11
x2
eNodeB
SGW
S5
PGW
SGi
Local
PDN
(Internet, Services, Apps)
UE
E-UTRAN
PCRF
EPC
User Plane
Control Plane
Home-PLMN
Home EPC
Home

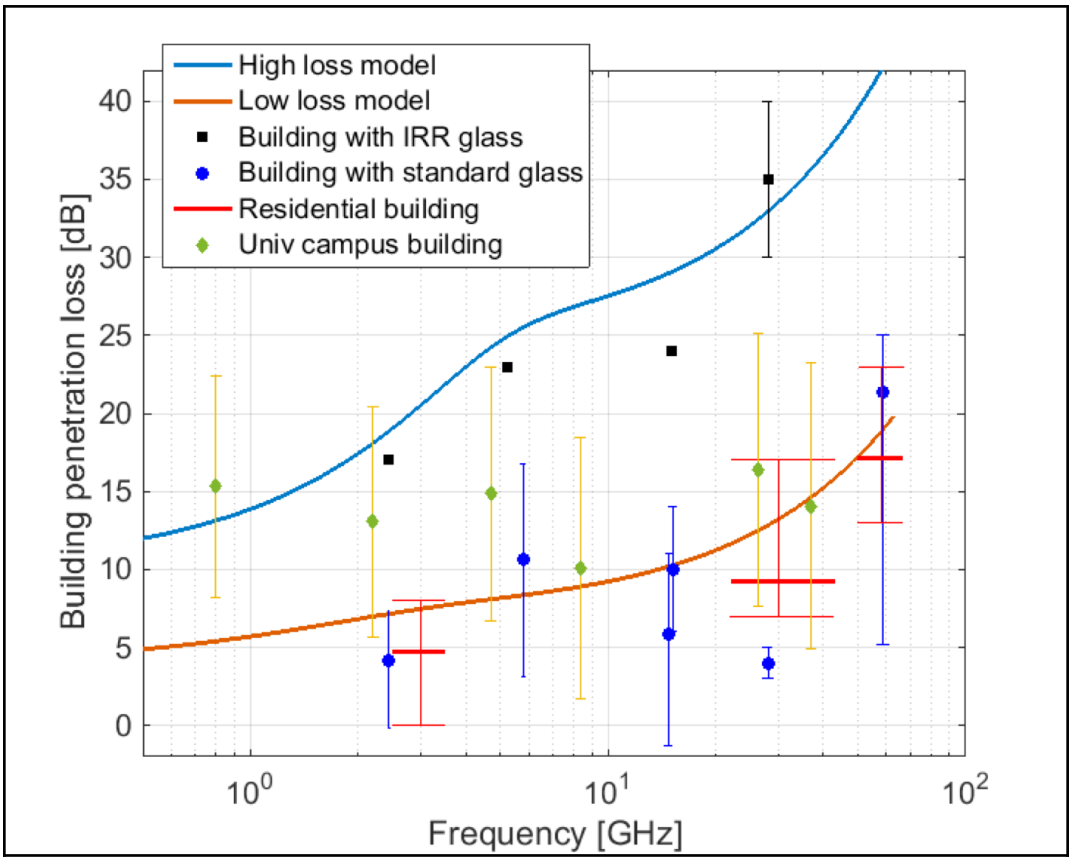

E-UTRAN Protocol Stack
Simplified OSI Model
Non Access Stratum (NAS)
Internet Protocol (IP)
5. Application Layer
4. Transport Layer
Control Plane Data
Radio Resource Control (RRC)
User Plane Data
3. Network Layer
PDCP Control
Packet Data Convergence Control (PDCP)
Radio Bearers
RLC Control
Radio Link Control (RLC)
Logical Channels
2. Data Link Layer
Configuration and Measurement
MAC Control
MAC (Medium Access Control)
Transport Channels
Physical Layer
Downlink: OFDM
Uplink: SC-OFDM
1. Physical Layer


Normal Traffic Behavior
Handover Setup
Handover
Completion
Normal Traffic Behavior
Internet
MME
SGW
Source
Destination
ENodeB
UE
S1 Bearer
Data Radio Bearer
X2 Interface
X2 Bearer
S1 Interface
S11 Interface

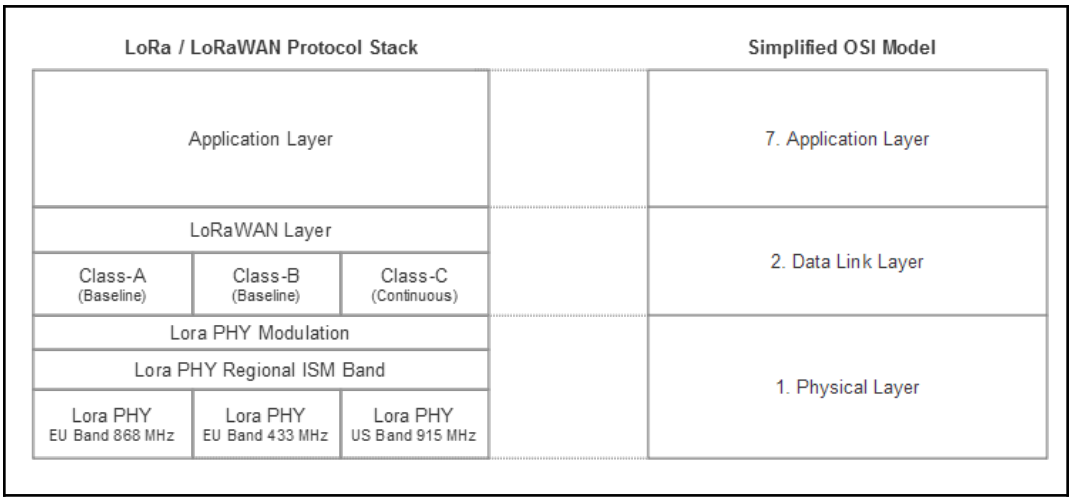

4G-LTE PHY Packet
SubFrame
Slot 0
Slot 2
Transport Block
Slot 19
4G-LTE MAC Data Frame
MAC Header
MAC SDU
Padding
4G-LTE RLC
RLC Header
RLC SDU
4G-LTE PDCP
PDCP Header
PDCP SDU
IP Packets

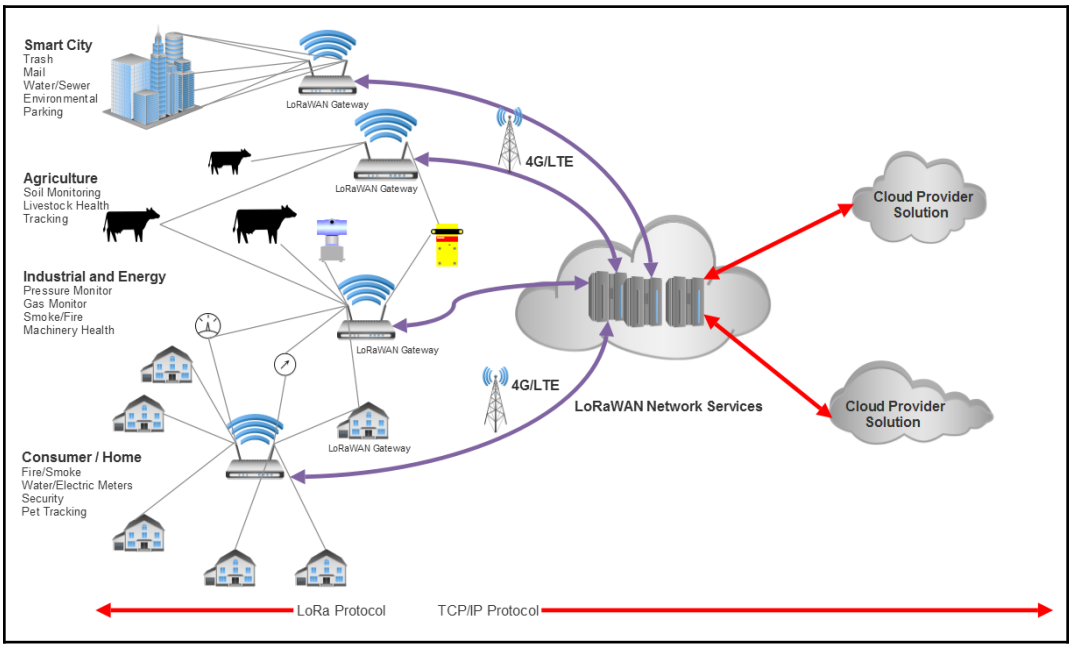

180 kHz Cat-NB Channel Residing in Guard Band
LTE Channel
LTE Channel
Guard Band Option
180 kHz Cat-NB Channel
GSM Option
180 kHz Cat-NB Channel
LTE Channel
In-Band Option

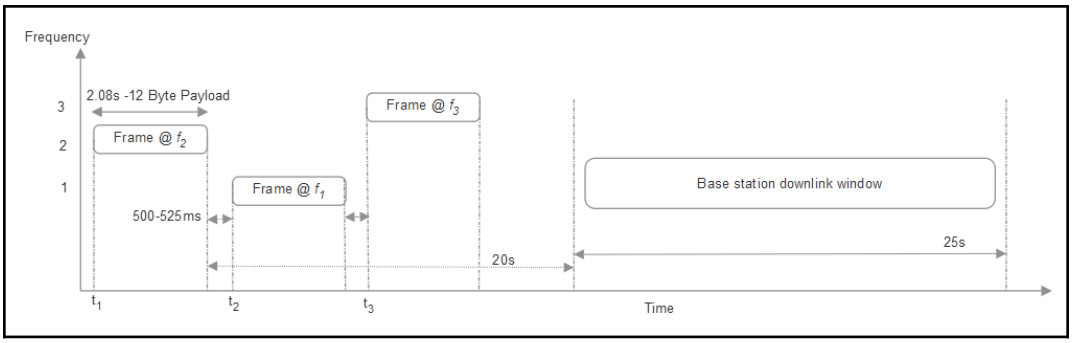

Residential and Indoor 5G Layering
Indoor 5G SmallCell 60GHz / WiFi
Indoor 5G SmallCell 60GHz / WiFi
5G 4GHz/ Cell
Forms of Device to Device Communication
Signaling and Data
5G UE
5G UE
5G 4GHz Mcrocell
D2D
5G UE
5G UE
Control
Data
30 GHz Small Cell
30 GHz Small Cell
30 GHz Small Cell
30 GHz Small Cell
Dense Small Cell mmWave
NextGen Core Backhaul
5G 4GHz/30GHzMacro Cell
1 Million IoT devices/km2
Dense Sensor and IoT deployments with various cell systems blanketing deployment
Inter-site Transport Link
Control
Data
5G 60GHz Small Cell
5G 4GHz Macro Cell Anchor
5G UE
Dual Connectivity
5G Macro Cell Massive MIMO Base Station
Massive MIMO with Beamforming

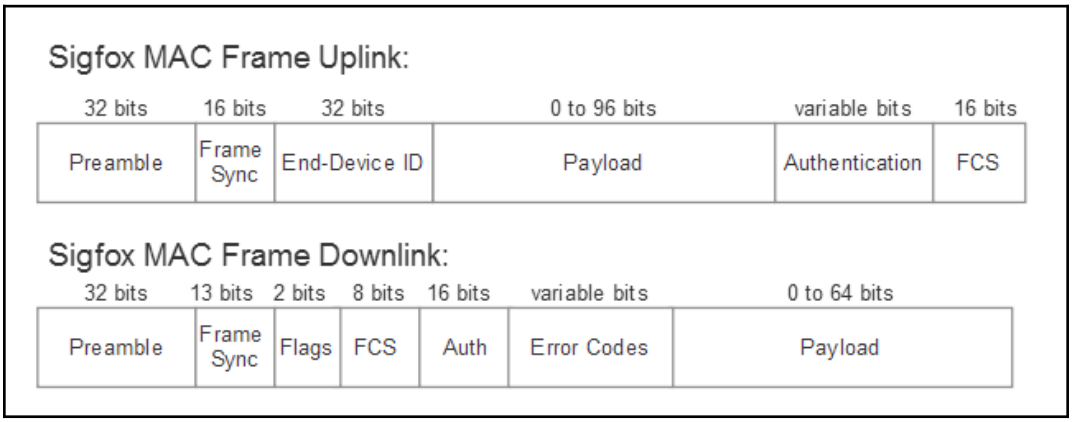


$$R_b = S \times \frac{1}{\left[\frac{2^S}{B}\right]}$$


LoRa / LoRaWAN Protocol Stack
Simplified OSI Model
Application Layer
7. Application Layer
LoRaWAN Layer
Class-A (Baseline)
Class-B (Baseline)
Class-C (Continuous)
2. Data Link Layer
Lora PHY Modulation
Lora PHY Regional ISM Band
Lora PHY EU Band 868 MHz
Lora PHY EU Band 433 MHz
Lora PHY US Band 915 MHz
1. Physical Layer


Smart City
Trash
Mail
Water/Sewer
Environmental
Parking
Agriculture
Soil Monitoring
Livestock Health
Tracking
Industrial and Energy
Pressure Monitor
Gas Monitor
Smoke/Fire
Machinery Health
Consumer / Home
Fire/Smoke
Water/Electric Meters
Security
Pet Tracking
LoRaWAN Gateway
LoRaWAN Gateway
LoRaWAN Gateway
LoRaWAN Gateway
4G/LTE
4G/LTE
LoRaWAN Network Services
Cloud Provider Solution
Cloud Provider Solution
LoRa Protocol
TCP/IP Protocol

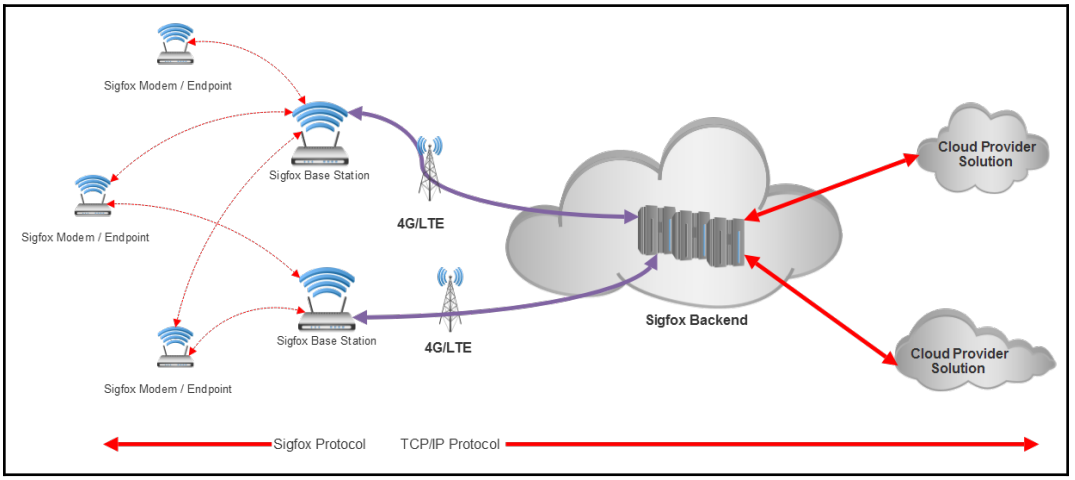




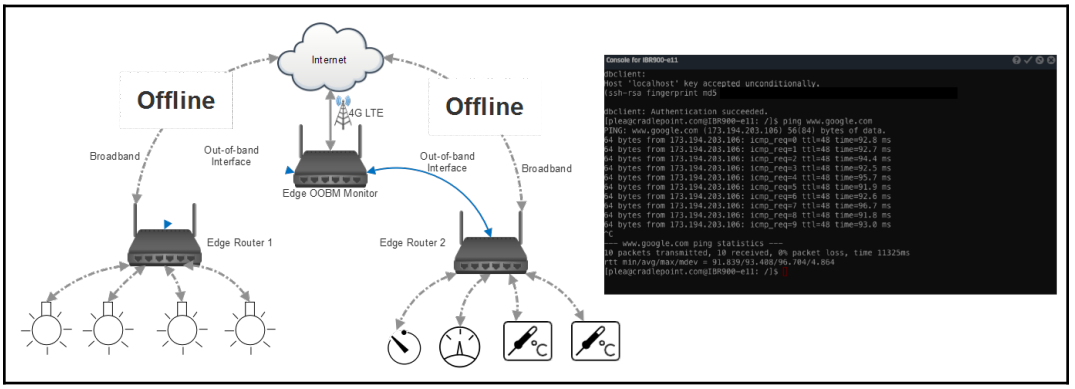
$$\frac{\sim 200\ bit\ uplink\ packet}{100bps} = 2seconds$$

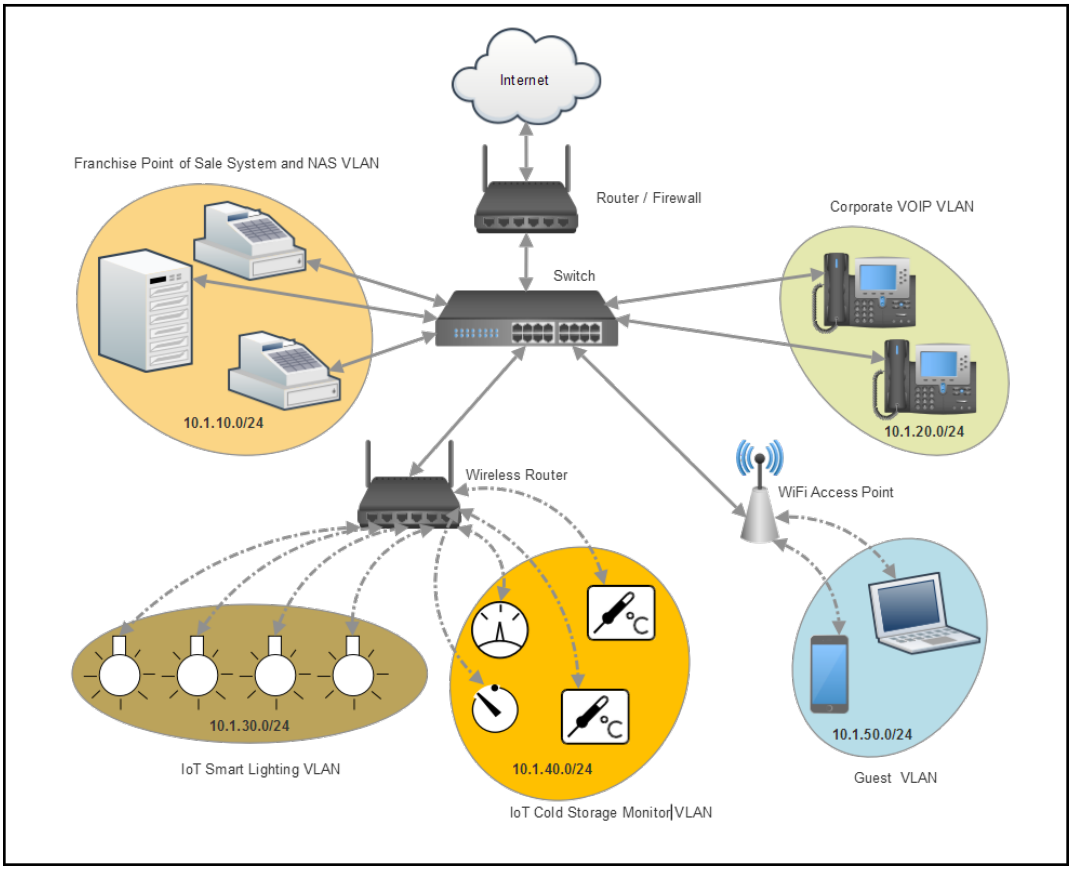
$$3600\ seconds@1\%\ duty\ cycle = \frac{36\ seconds\ message\ transmission\ time}{hour} \times \frac{message}{3\ repetitions \times 2\ seconds} = \frac{6\ messages}{hour}$$

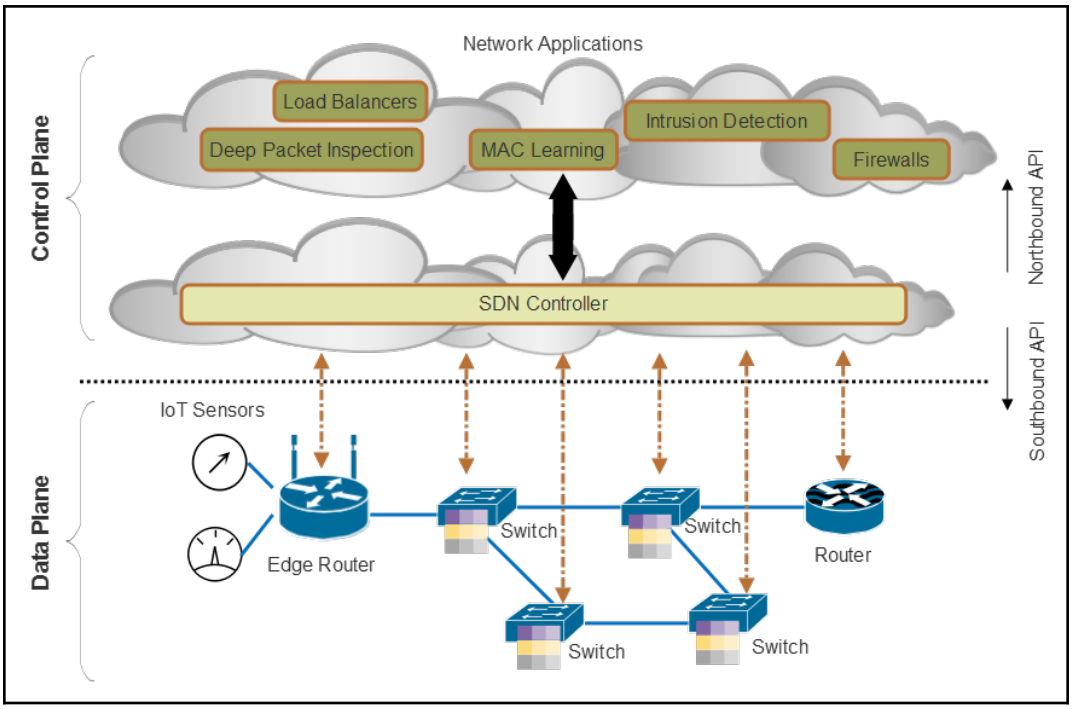
Sigfox Protocol Stack		Simplified OSI Model
Application Layer		7. Application Layer
		6. Presentation Layer
		5. Session Layer
Frame		4. Transport Layer
		3. Network Layer
MAC Layer		2. Data Link Layer
PHY Layer (868MHz / 902MHz Radios)		1. Physical Layer

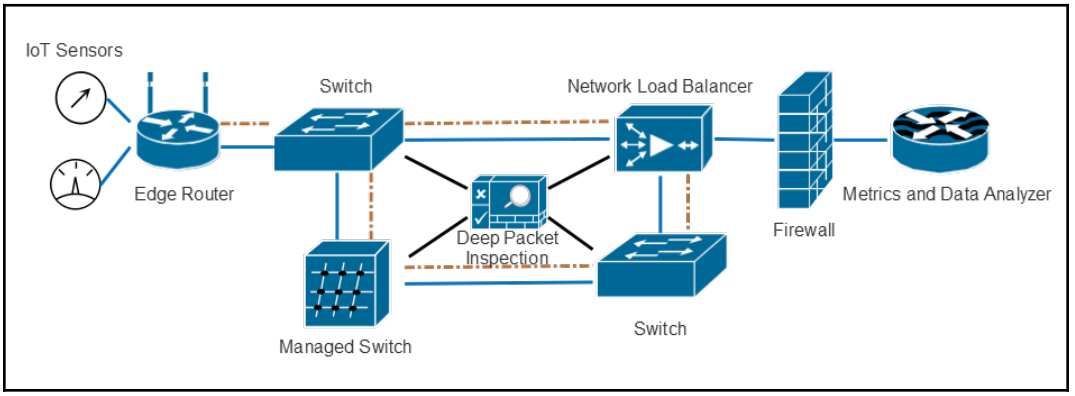


Chapter 8: Routers and Gateways

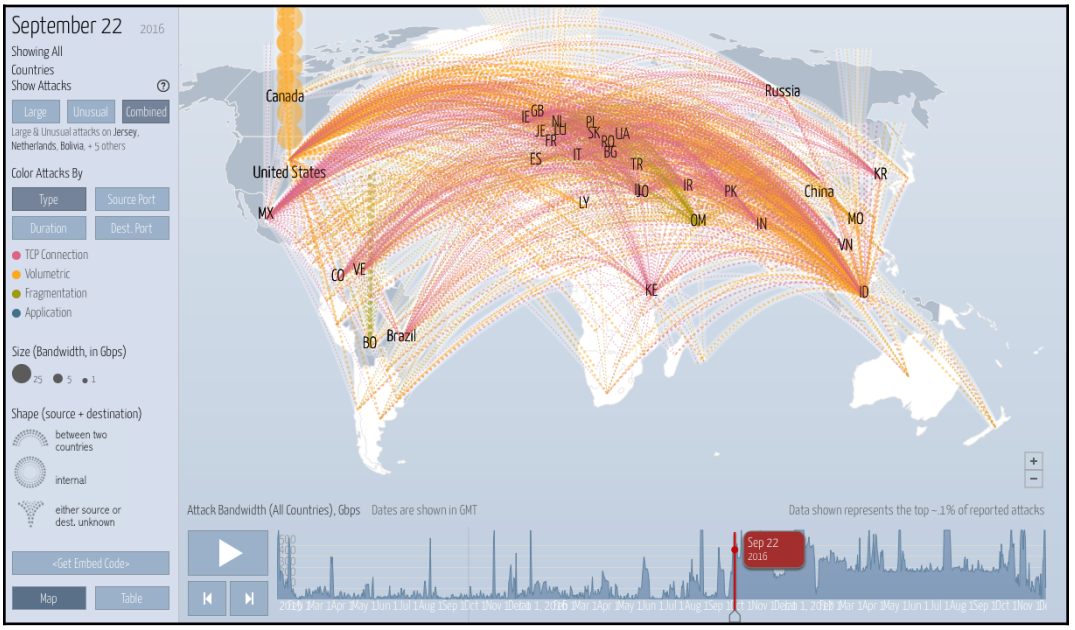


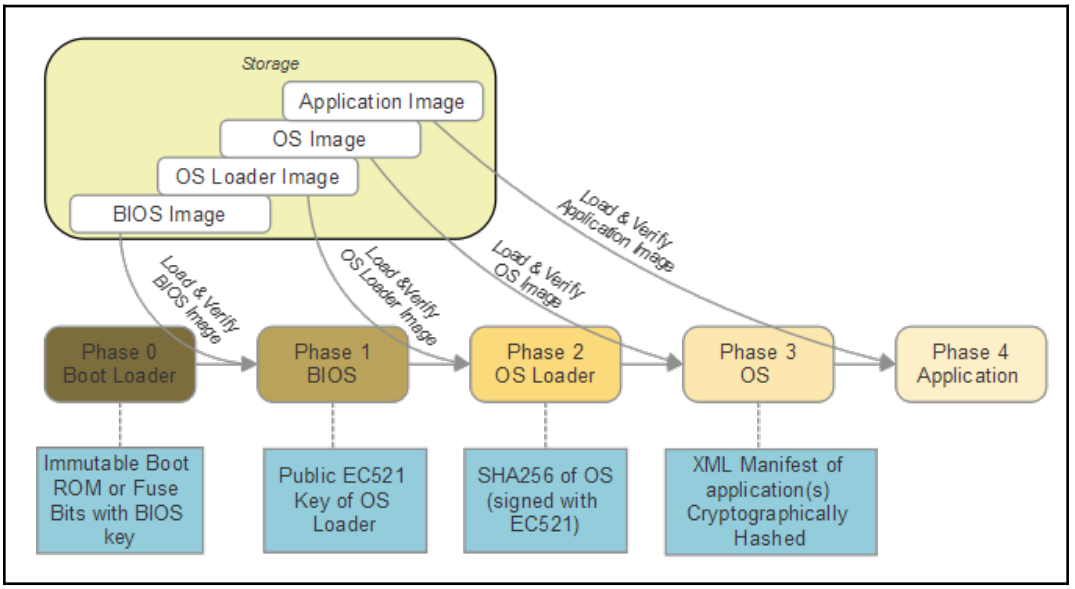


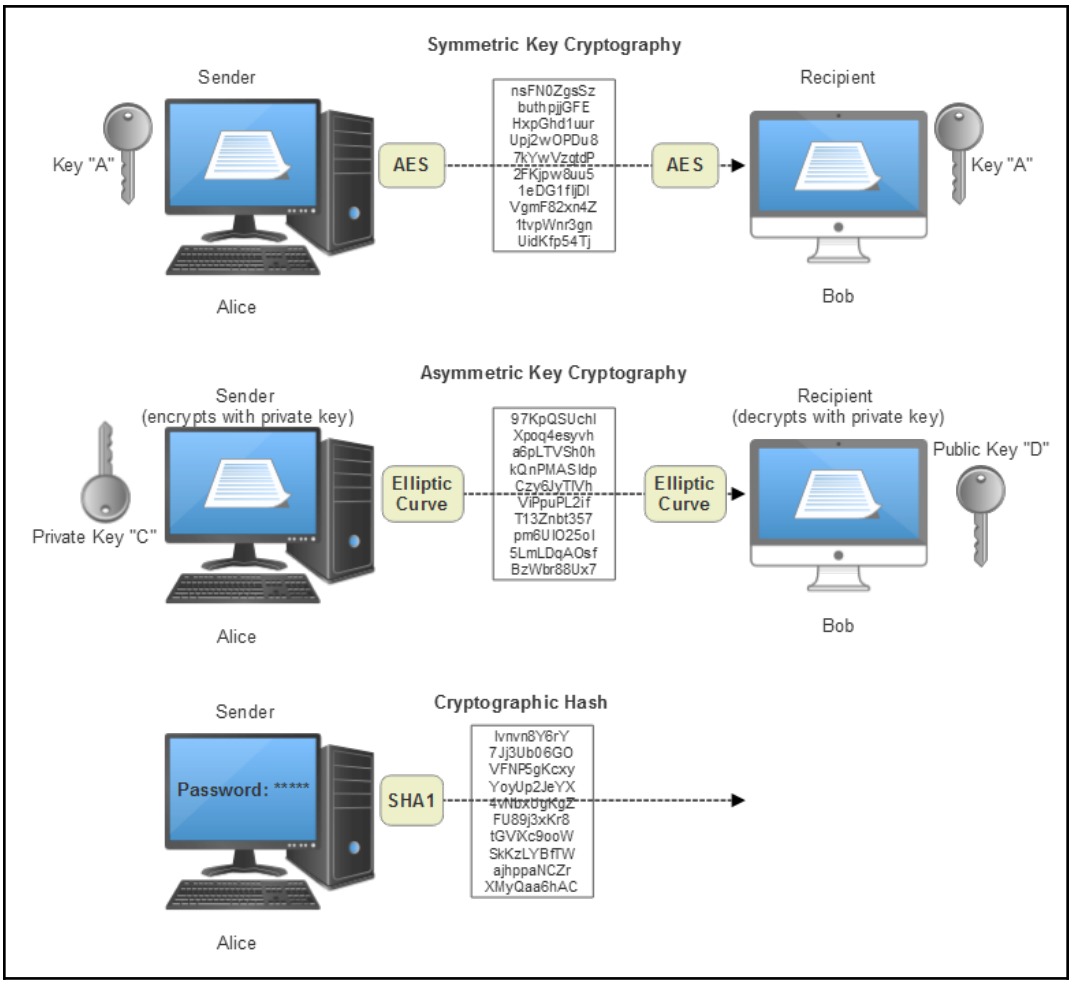

Network Applications
Load Balancers
Deep Packet Inspection
MAC Learning
Intrusion Detection
Firewalls
Control Plane
Northbound API
SDN Controller
Southbound API
IoT Sensors
Data Plane
Edge Router
Switch
Switch
Router
Switch
Switch

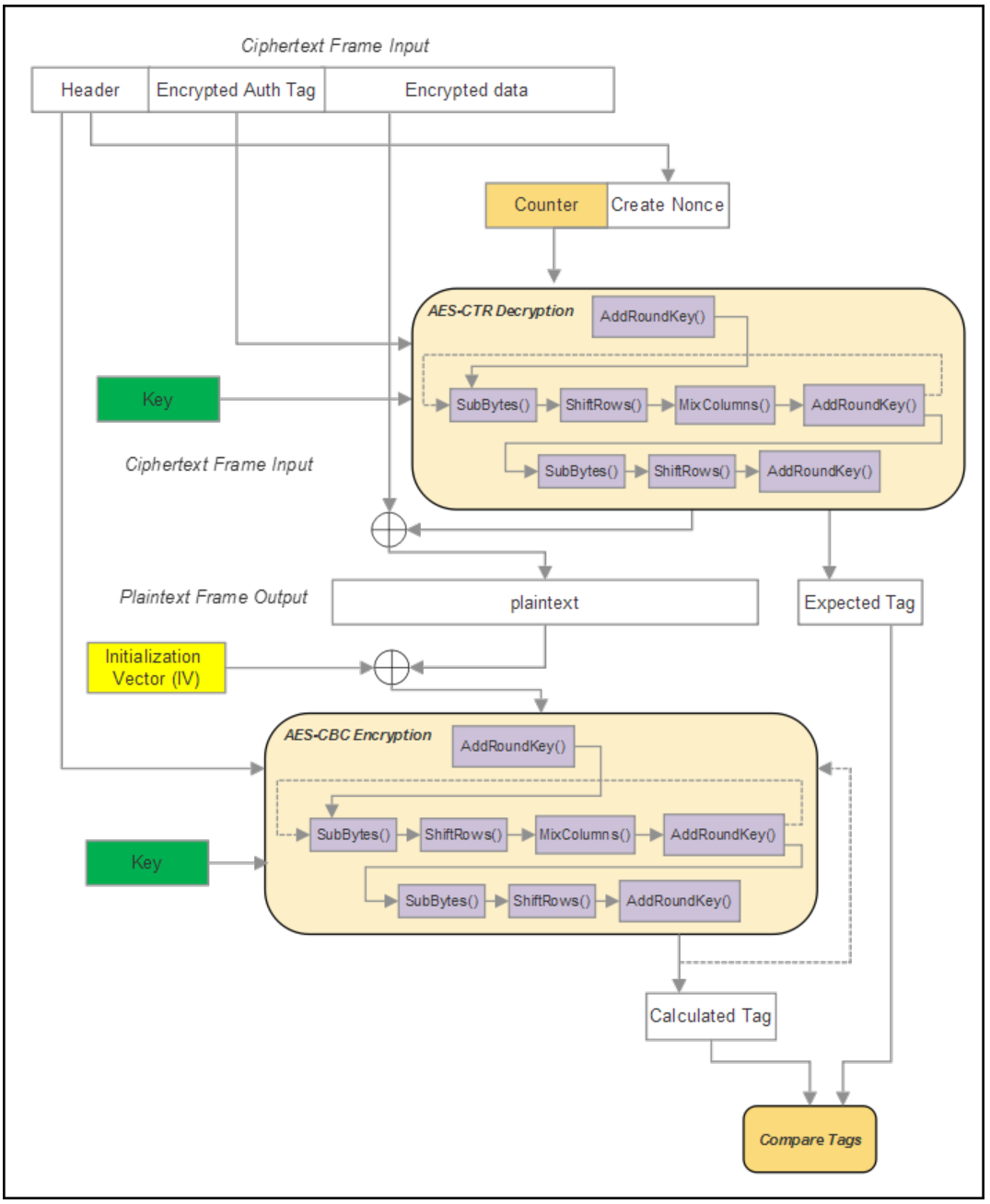

IoT Sensors
Edge Router
Switch
Network Load Balancer
Deep Packet Inspection
Firewall
Metrics and Data Analyzer
Managed Switch
Switch

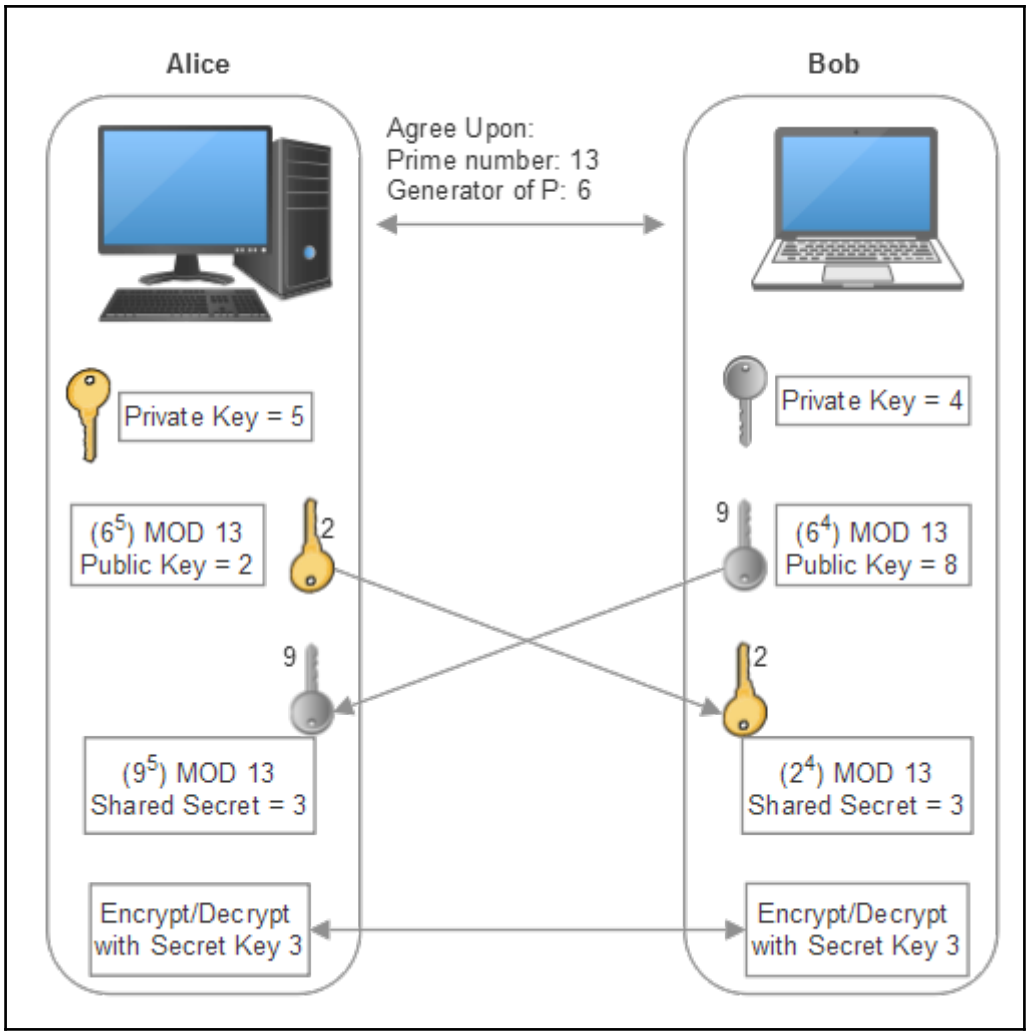
Chapter 9: IoT Security

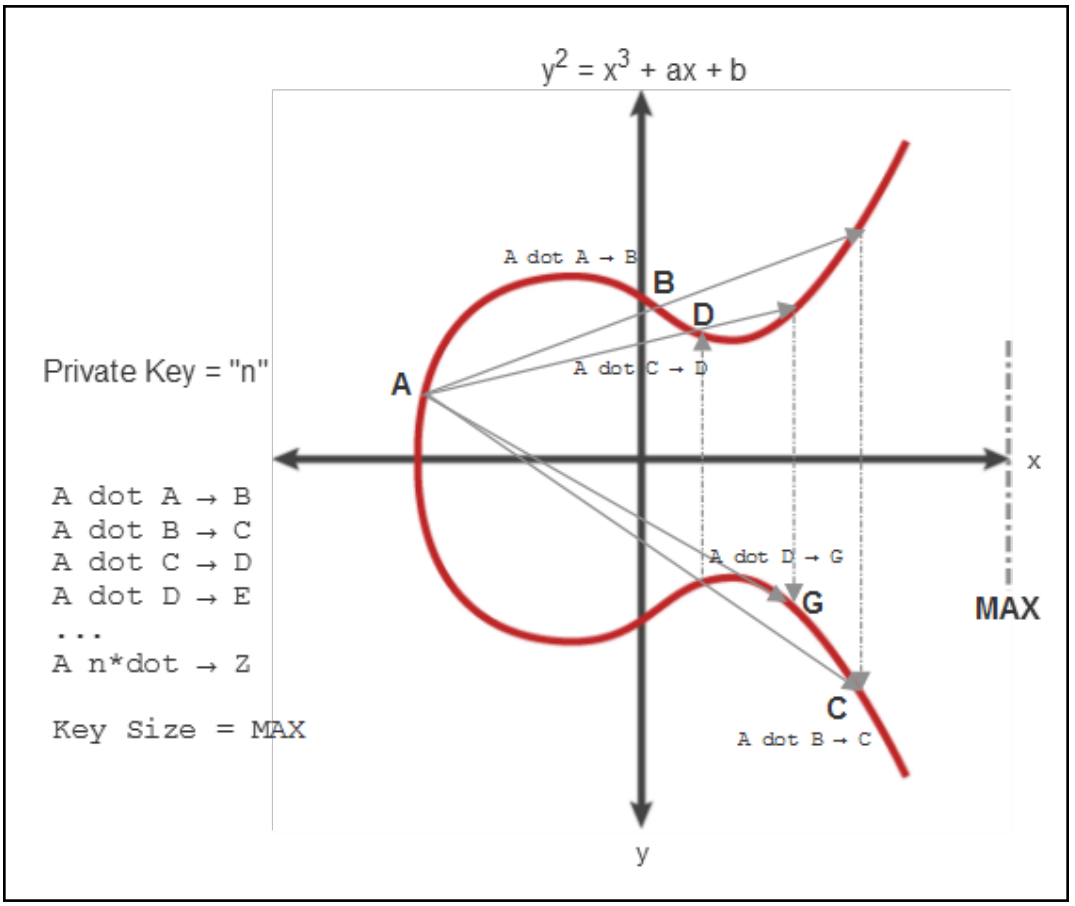


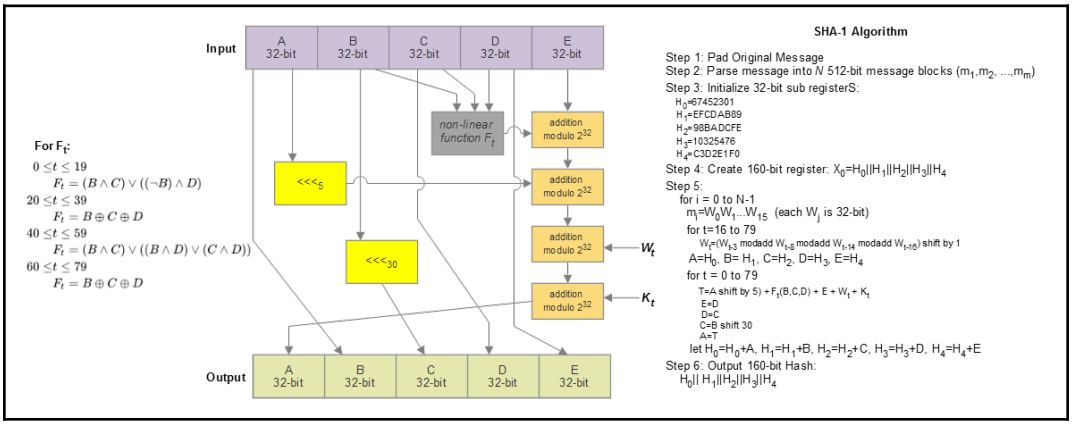


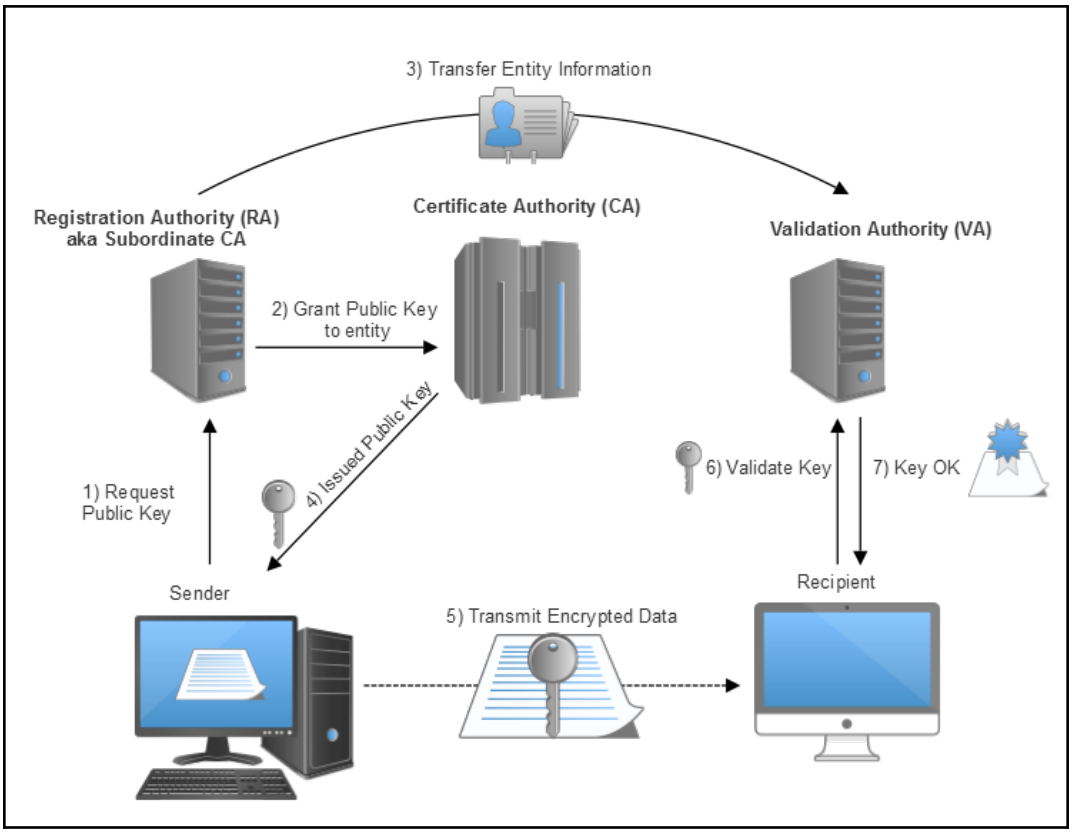

Symmetric Key Cryptography
Sender
Key "A"
AES
nsFN0ZgsSz
buthpjjGFE
HxpGhd1uur
Upj2wOPDu8
7kYwVzqtdP
2FKjpw8uu5
1eDG1fIjDl
VgmF82xn4Z
1tvpWnr3gn
UidKfp54Tj
AES
Recipient
Key "A"
Alice
Bob
Asymmetric Key Cryptography
Sender
(encrypts with private key)
Private Key "C"
Elliptic Curve
97KpQSUchl
Xpoq4esyvh
a6pLTVSh0h
kQnPMASldp
Czy6JyTlVh
ViPpuPL2if
T13Znbt357
pm6UlO25oI
5LmLDqAOsf
BzWbr88Ux7
Elliptic Curve
Recipient
(decrypts with private key)
Public Key "D"
Alice
Bob
Cryptographic Hash
Sender
Password: *****
SHA1
lvnvn8Y6rY
7Jj3Ub06GO
VFNP5gKcxy
YoyUp2JeYX
4vNbxDgKgZ
FU89j3xKr8
tGViKc9ooW
SkKzLYBfTW
ajhppaNCZr
XMyQaa6hAC
Alice

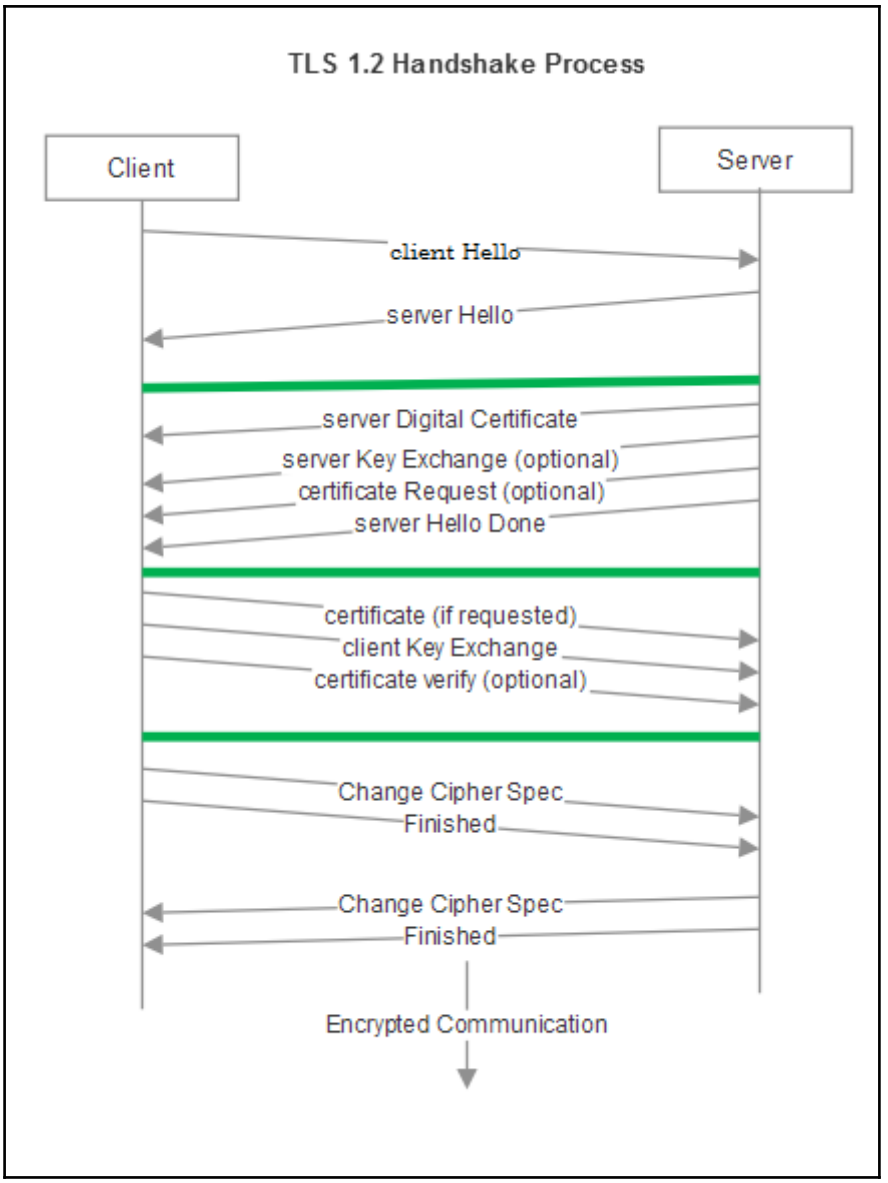

Ciphertext Frame Input
Header
Encrypted Auth Tag
Encrypted data
Counter
Create Nonce
AES-CTR Decryption
AddRoundKey()
Key
SubBytes()
ShiftRows()
MixColumns()
AddRoundKey()
SubBytes()
ShiftRows()
AddRoundKey()
Ciphertext Frame Input
Plaintext Frame Output
plaintext
Expected Tag
Initialization Vector (IV)
AES-CBC Encryption
AddRoundKey()
Key
SubBytes()
ShiftRows()
MixColumns()
AddRoundKey()
SubBytes()
ShiftRows()
AddRoundKey()
Calculated Tag
Compare Tags

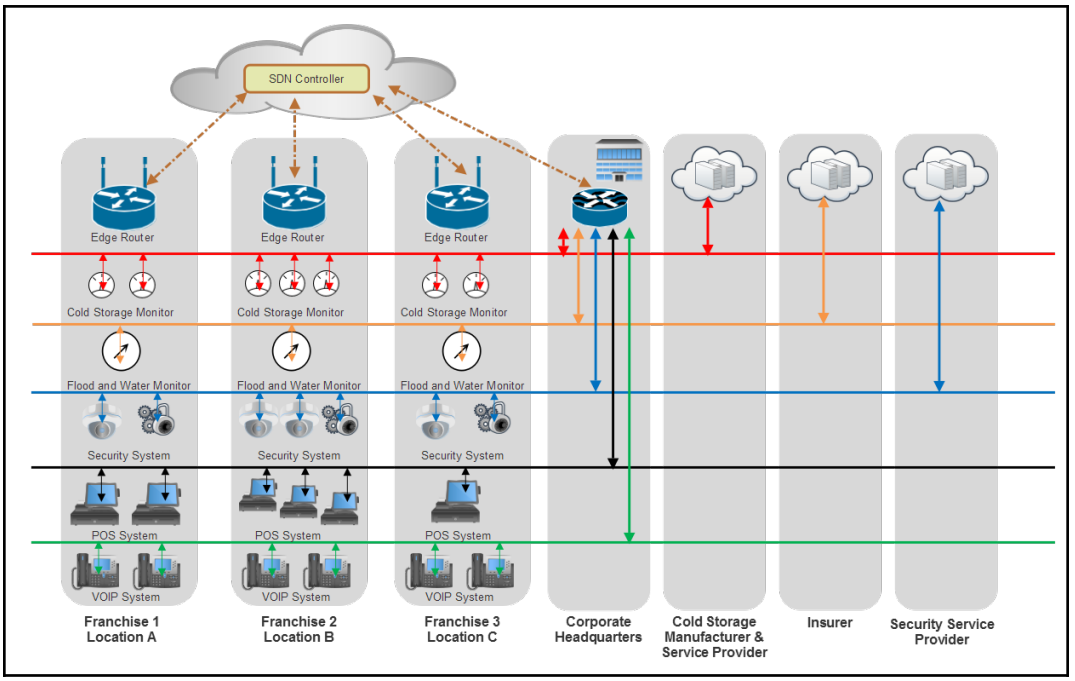

Alice
Bob
Agree Upon:
Prime number: 13
Generator of P: 6
Private Key = 5
Private Key = 4
(6^5) MOD 13
Public Key = 2
2
9
(6^4) MOD 13
Public Key = 8
9
2
(9^5) MOD 13
Shared Secret = 3
(2^4) MOD 13
Shared Secret = 3
Encrypt/Decrypt
with Secret Key 3
Encrypt/Decrypt
with Secret Key 3

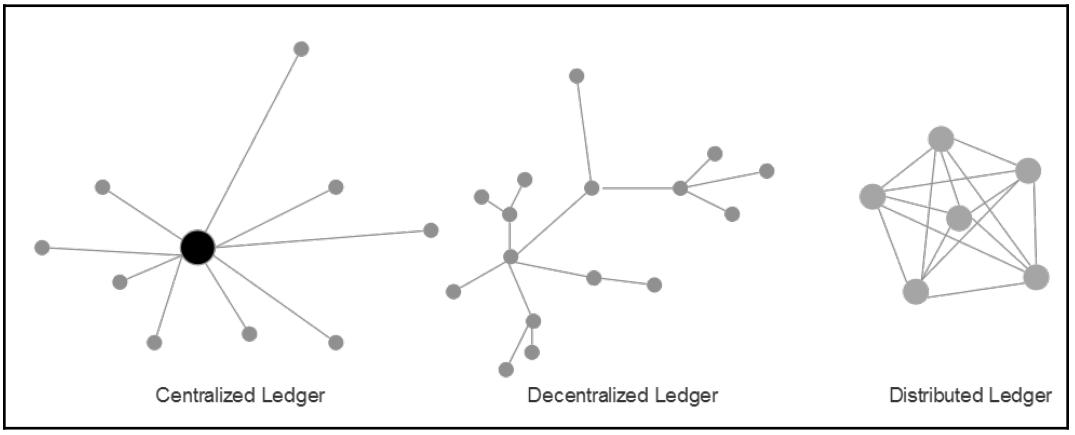

$y^2 = x^3 + ax + b$
Private Key = "n"
A dot A → B
A dot B → C
A dot C → D
A dot D → E
...
A n*dot → Z
Key Size = MAX
A dot A → B
A dot C → D
A dot D → G
A dot B → C
A
B
C
D
G
MAX
x
y

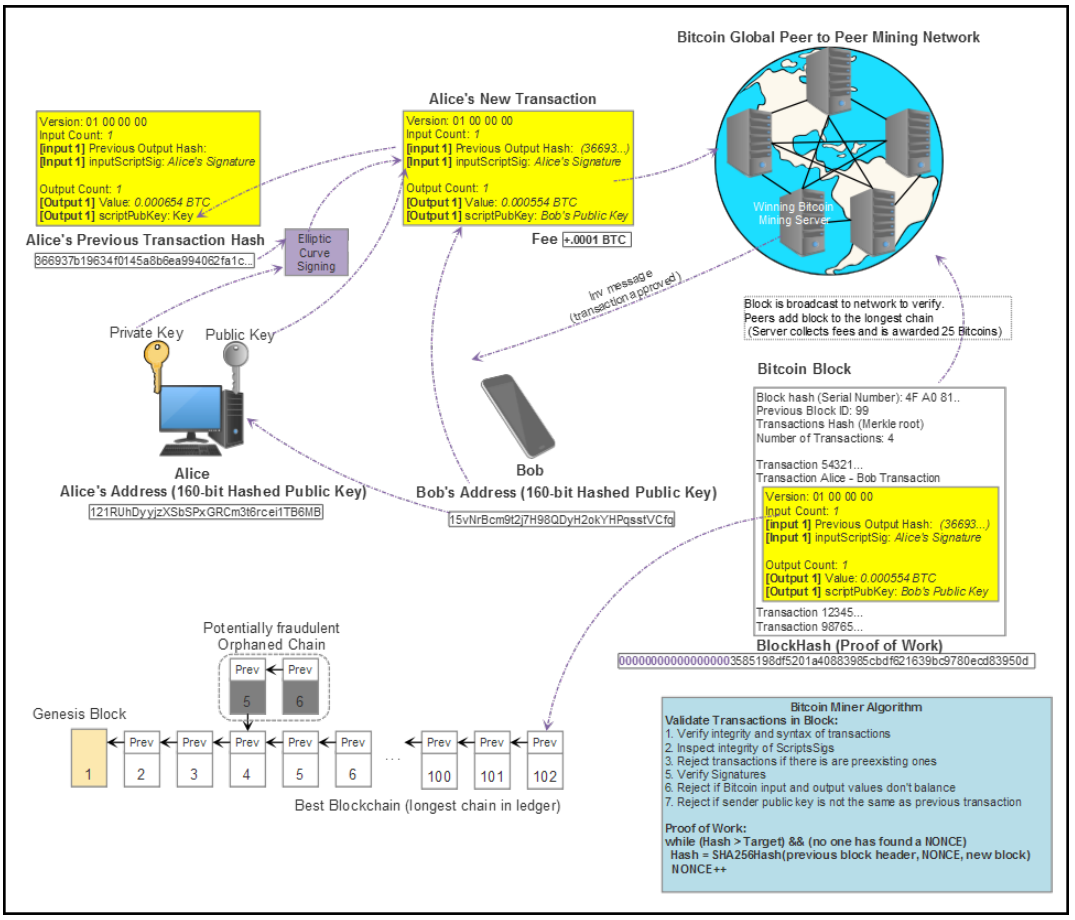

Input
A 32-bit
B 32-bit
C 32-bit
D 32-bit
E 32-bit
non-linear function F_t
addition modulo 2^{32}
<<<5
<<<30
W_t
K_t
Output
For F_t:
$0 \le t \le 19$
$F_t = (B \wedge C) \vee ((\neg B) \wedge D)$
$20 \le t \le 39$
$F_t = B \oplus C \oplus D$
$40 \le t \le 59$
$F_t = (B \wedge C) \vee ((B \wedge D) \vee (C \wedge D))$
$60 \le t \le 79$
$F_t = B \oplus C \oplus D$
SHA-1 Algorithm
Step 1: Pad Original Message
Step 2: Parse message into N 512-bit message blocks ($m_1, m_2, \ldots, m_n$)
Step 3: Initialize 32-bit sub registerS:
H_0=67452301
H_1=EFCDAB89
H_2=98BADCFE
H_3=10325476
H_4=C3D2E1F0
Step 4: Create 160-bit register: $X_0=H_0||H_1||H_2||H_3||H_4$
Step 5:
for i = 0 to N-1
$m_i=W_0W_1...W_{15}$ (each W_j is 32-bit)
for t=16 to 79
W_t=(W_{t-3} modadd W_{t-8} modadd W_{t-14} modadd W_{t-16}) shift by 1
A=H_0, B= H_1, C=H_2, D=H_3, E=H_4
for t = 0 to 79
T=A shift by 5) + F_t(B,C,D) + E + W_t + K_t
E=D
D=C
C=B shift 30
A=T
let $H_0=H_0+A$, $H_1=H_1+B$, $H_2=H_2+C$, $H_3=H_3+D$, $H_4=H_4+E$
Step 6: Output 160-bit Hash:
$H_0||H_1||H_2||H_3||H_4$

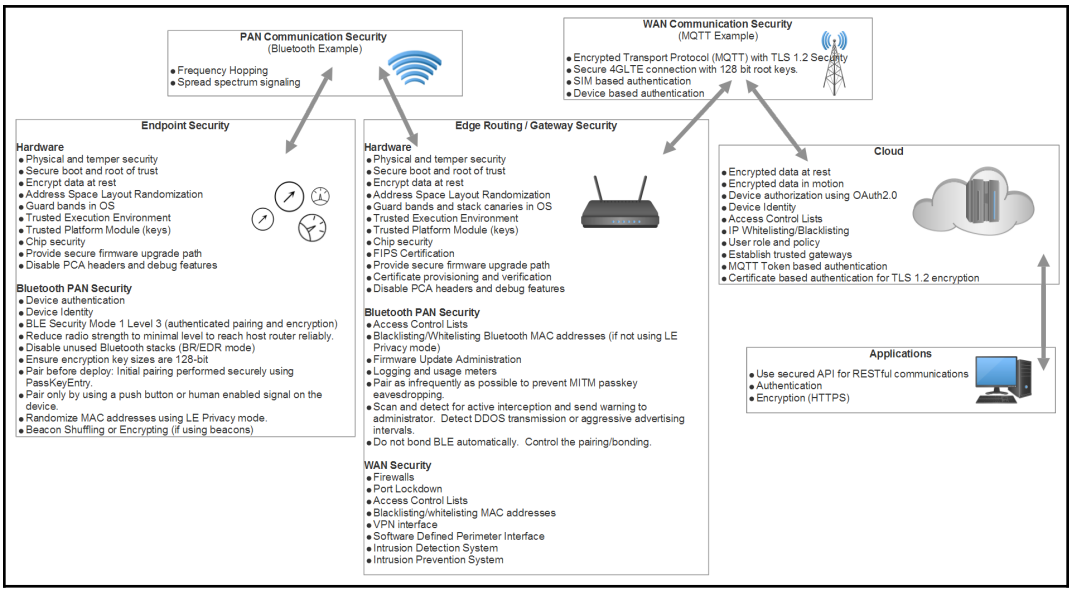

3) Transfer Entity Information
Registration Authority (RA)
aka Subordinate CA
Certificate Authority (CA)
Validation Authority (VA)
2) Grant Public Key
to entity
4) Issued Public Key
1) Request
Public Key
6) Validate Key
7) Key OK
Sender
Recipient
5) Transmit Encrypted Data


TLS 1.2 Handshake Process
Client
Server
client Hello
server Hello
server Digital Certificate
server Key Exchange (optional)
certificate Request (optional)
server Hello Done
certificate (if requested)
client Key Exchange
certificate verify (optional)
Change Cipher Spec
Finished
Change Cipher Spec
Finished
Encrypted Communication

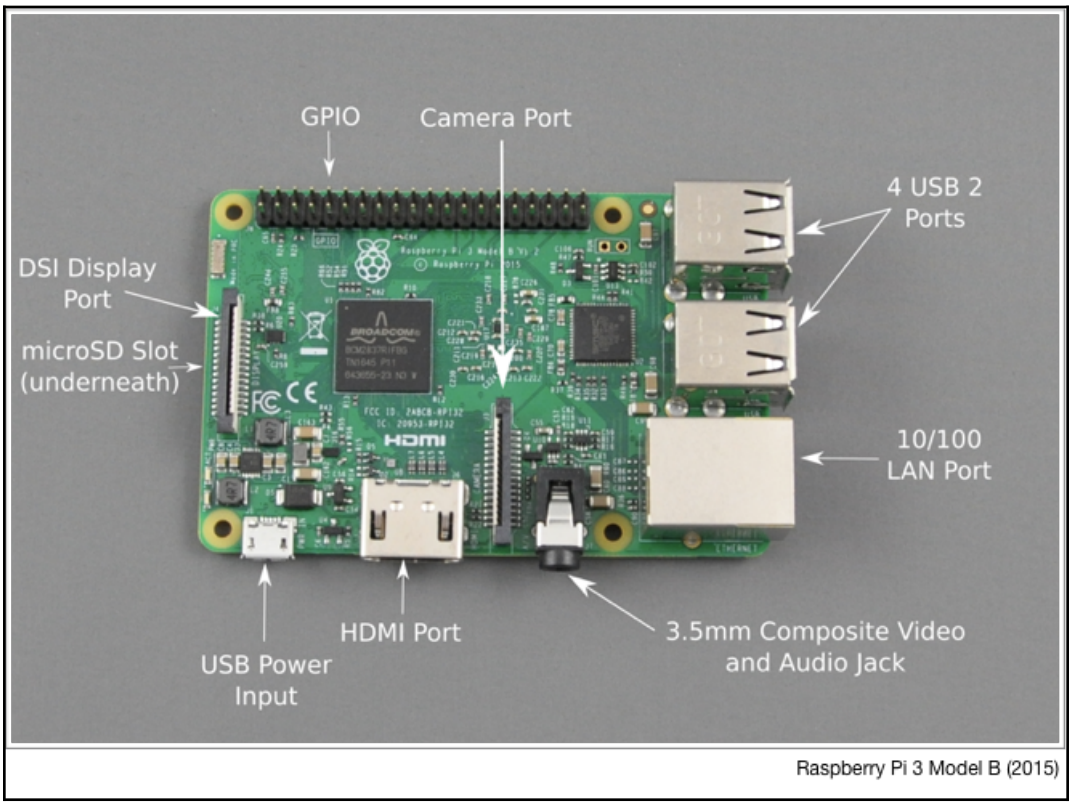

SDN Controller
Edge Router
Edge Router
Edge Router
Cold Storage Monitor
Cold Storage Monitor
Cold Storage Monitor
Flood and Water Monitor
Flood and Water Monitor
Flood and Water Monitor
Security System
Security System
Security System
POS System
POS System
POS System
VOIP System
VOIP System
VOIP System
Franchise 1 Location A
Franchise 2 Location B
Franchise 3 Location C
Corporate Headquarters
Cold Storage Manufacturer & Service Provider
Insurer
Security Service Provider


Centralized Ledger
Decentralized Ledger
Distributed Ledger

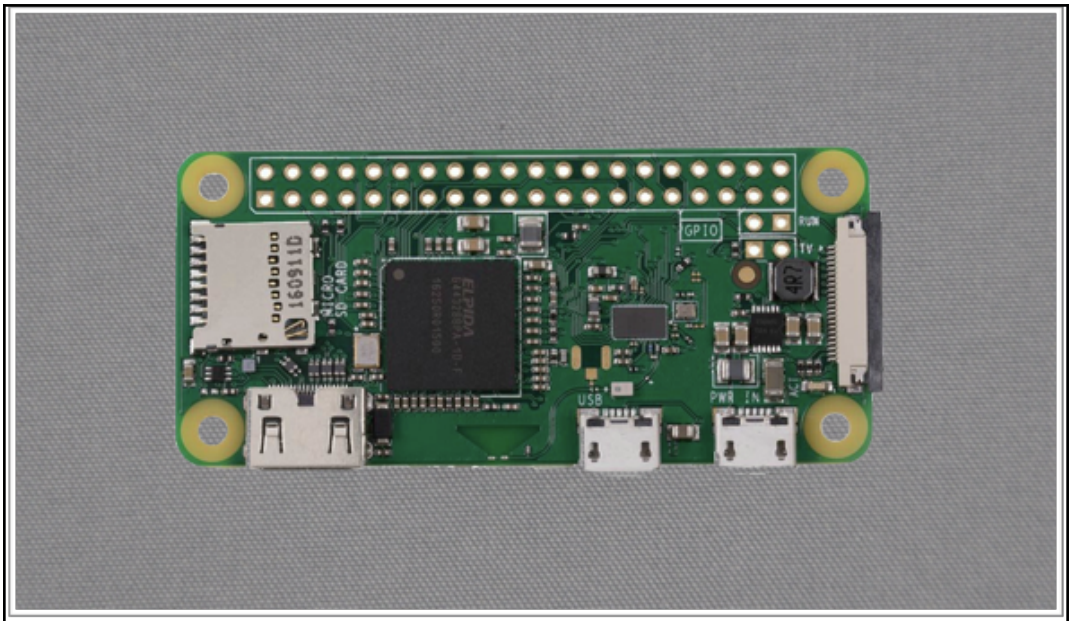

Bitcoin Global Peer to Peer Mining Network
Alice's New Transaction
Version: 01 00 00 00
Input Count: 1
[Input 1] Previous Output Hash:
[Input 1] inputScriptSig: Alice's Signature
Output Count: 1
[Output 1] Value: 0.000654 BTC
[Output 1] scriptPubKey: Key
Alice's Previous Transaction Hash
366937b19634f0145a8b6ea994062fa1c...
Elliptic Curve Signing
Version: 01 00 00 00
Input Count: 1
[Input 1] Previous Output Hash: (36693...)
[Input 1] inputScriptSig: Alice's Signature
Output Count: 1
[Output 1] Value: 0.000554 BTC
[Output 1] scriptPubKey: Bob's Public Key
Fee +.0001 BTC
Winning Bitcoin Mining Server
Inv message (transaction approved)
Block is broadcast to network to verify.
Peers add block to the longest chain
(Server collects fees and is awarded 25 Bitcoins)
Private Key
Public Key
Alice
Alice's Address (160-bit Hashed Public Key)
121RUhDyyjzXSbSPxGRCm3t8rcei1TB6MB
Bob
Bob's Address (160-bit Hashed Public Key)
15vNrBcm9t2j7H98QDyH2okYHPqsstVCfg
Bitcoin Block
Block hash (Serial Number): 4F A0 81..
Previous Block ID: 99
Transactions Hash (Merkle root)
Number of Transactions: 4
Transaction 54321...
Transaction Alice - Bob Transaction
Version: 01 00 00 00
Input Count: 1
[Input 1] Previous Output Hash: (36693...)
[Input 1] inputScriptSig: Alice's Signature
Output Count: 1
[Output 1] Value: 0.000554 BTC
[Output 1] scriptPubKey: Bob's Public Key
Transaction 12345...
Transaction 98765...
BlockHash (Proof of Work)
00000000000000000003585198df5201a40883985cbdf621639bc9780ecd83950d
Potentially fraudulent Orphaned Chain
Prev
5
6
Genesis Block
1
2
3
4
5
6
100
101
102
Best Blockchain (longest chain in ledger)
Bitcoin Miner Algorithm
Validate Transactions in Block:
1. Verify integrity and syntax of transactions
2. Inspect integrity of ScriptsSigs
3. Reject transactions if there is are preexisting ones
5. Verify Signatures
6. Reject if Bitcoin input and output values don't balance
7. Reject if sender public key is not the same as previous transaction
Proof of Work:
while (Hash > Target) && (no one has found a NONCE)
Hash = SHA256Hash(previous block header, NONCE, new block)
NONCE++


PAN Communication Security
(Bluetooth Example)
Frequency Hopping
Spread spectrum signaling
WAN Communication Security
(MQTT Example)
Encrypted Transport Protocol (MQTT) with TLS 1.2 Security
Secure 4GLTE connection with 128 bit root keys.
SIM based authentication
Device based authentication
Endpoint Security
Hardware
Physical and temper security
Secure boot and root of trust
Encrypt data at rest
Address Space Layout Randomization
Guard bands in OS
Trusted Execution Environment
Trusted Platform Module (keys)
Chip security
Provide secure firmware upgrade path
Disable PCA headers and debug features
Bluetooth PAN Security
Device authentication
Device Identity
BLE Security Mode 1 Level 3 (authenticated pairing and encryption)
Reduce radio strength to minimal level to reach host router reliably.
Disable unused Bluetooth stacks (BR/EDR mode)
Ensure encryption key sizes are 128-bit
Pair before deploy: Initial pairing performed securely using PassKeyEntry.
Pair only by using a push button or human enabled signal on the device.
Randomize MAC addresses using LE Privacy mode.
Beacon Shuffling or Encrypting (if using beacons)
Edge Routing / Gateway Security
Hardware
Physical and temper security
Secure boot and root of trust
Encrypt data at rest
Address Space Layout Randomization
Guard bands and stack canaries in OS
Trusted Execution Environment
Trusted Platform Module (keys)
Chip security
FIPS Certification
Provide secure firmware upgrade path
Certificate provisioning and verification
Disable PCA headers and debug features
Bluetooth PAN Security
Access Control Lists
Blacklisting/Whitelisting Bluetooth MAC addresses (if not using LE Privacy mode)
Firmware Update Administration
Logging and usage meters
Pair as infrequently as possible to prevent MITM passkey eavesdropping.
Scan and detect for active interception and send warning to administrator. Detect DDOS transmission or aggressive advertising intervals.
Do not bond BLE automatically. Control the pairing/bonding.
WAN Security
Firewalls
Port Lockdown
Access Control Lists
Blacklisting/whitelisting MAC addresses
VPN interface
Software Defined Perimeter Interface
Intrusion Detection System
Intrusion Prevention System
Cloud
Encrypted data at rest
Encrypted data in motion
Device authorization using OAuth2.0
Device Identity
Access Control Lists
IP Whitelisting/Blacklisting
User role and policy
Establish trusted gateways
MQTT Token based authentication
Certificate based authentication for TLS 1.2 encryption
Applications
Use secured API for RESTful communications
Authentication
Encryption (HTTPS)

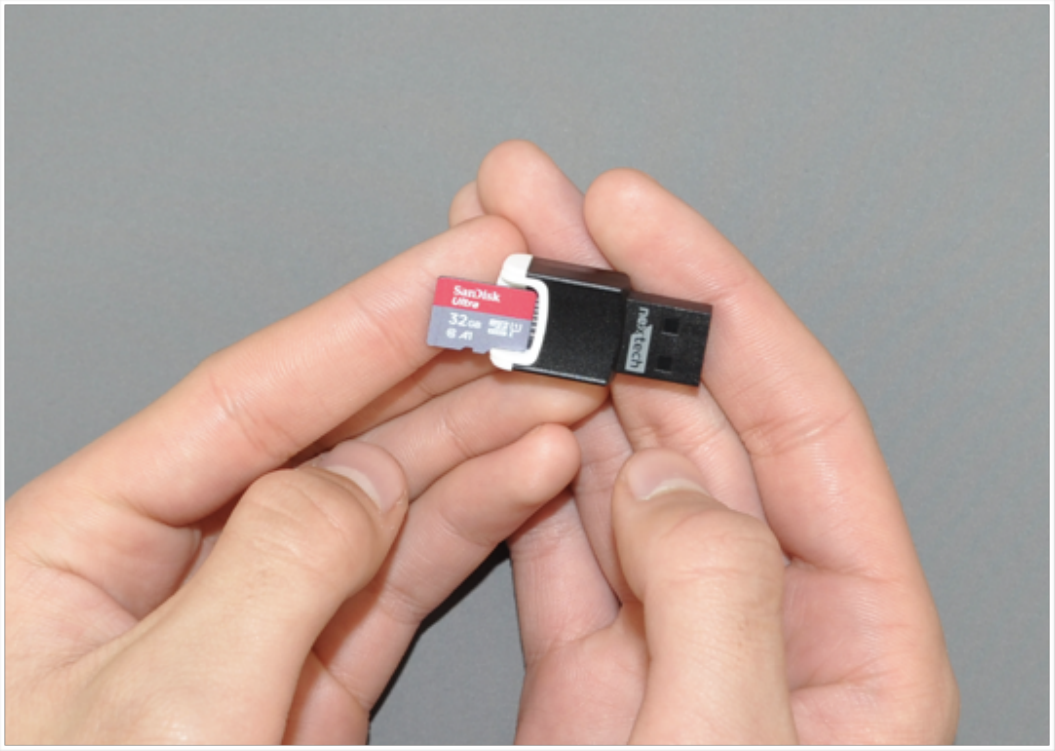
Chapter 10: Installing Raspbian on the Raspberry Pi



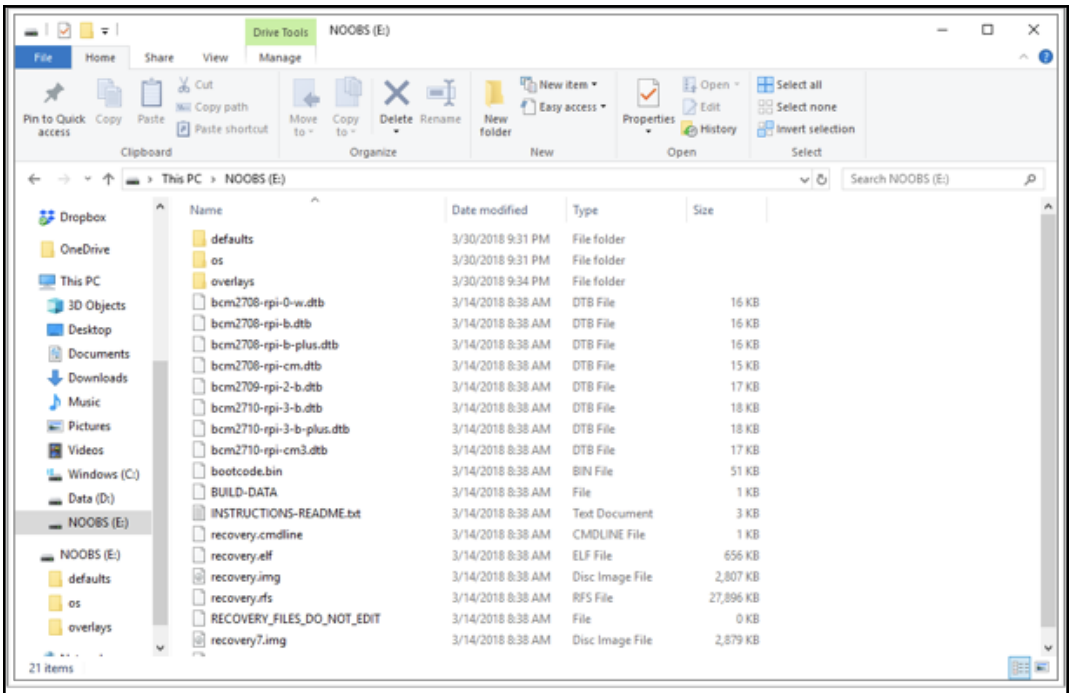
Raspberry Pi 3 Model B (2015)



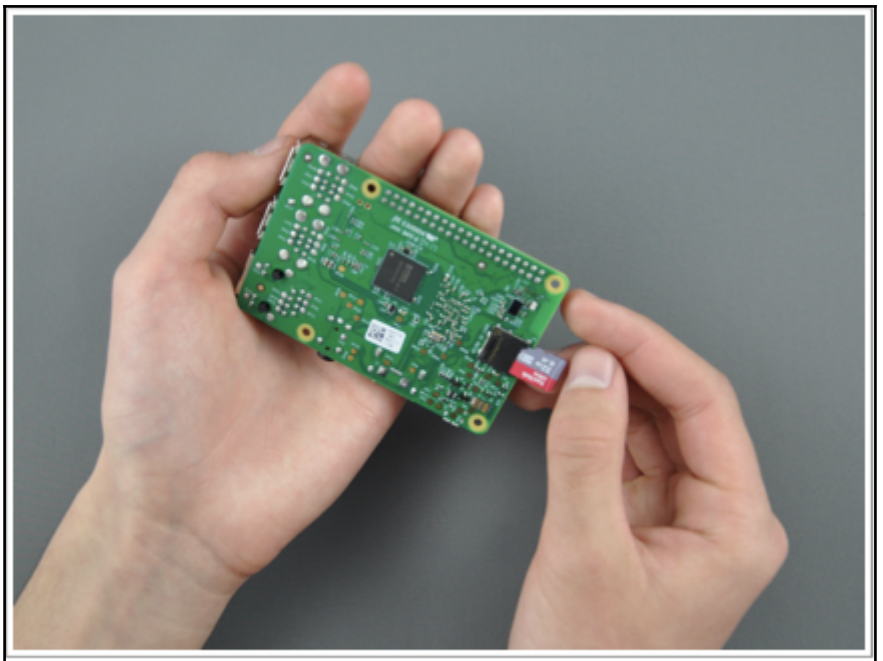
Raspberry Pi Zero W



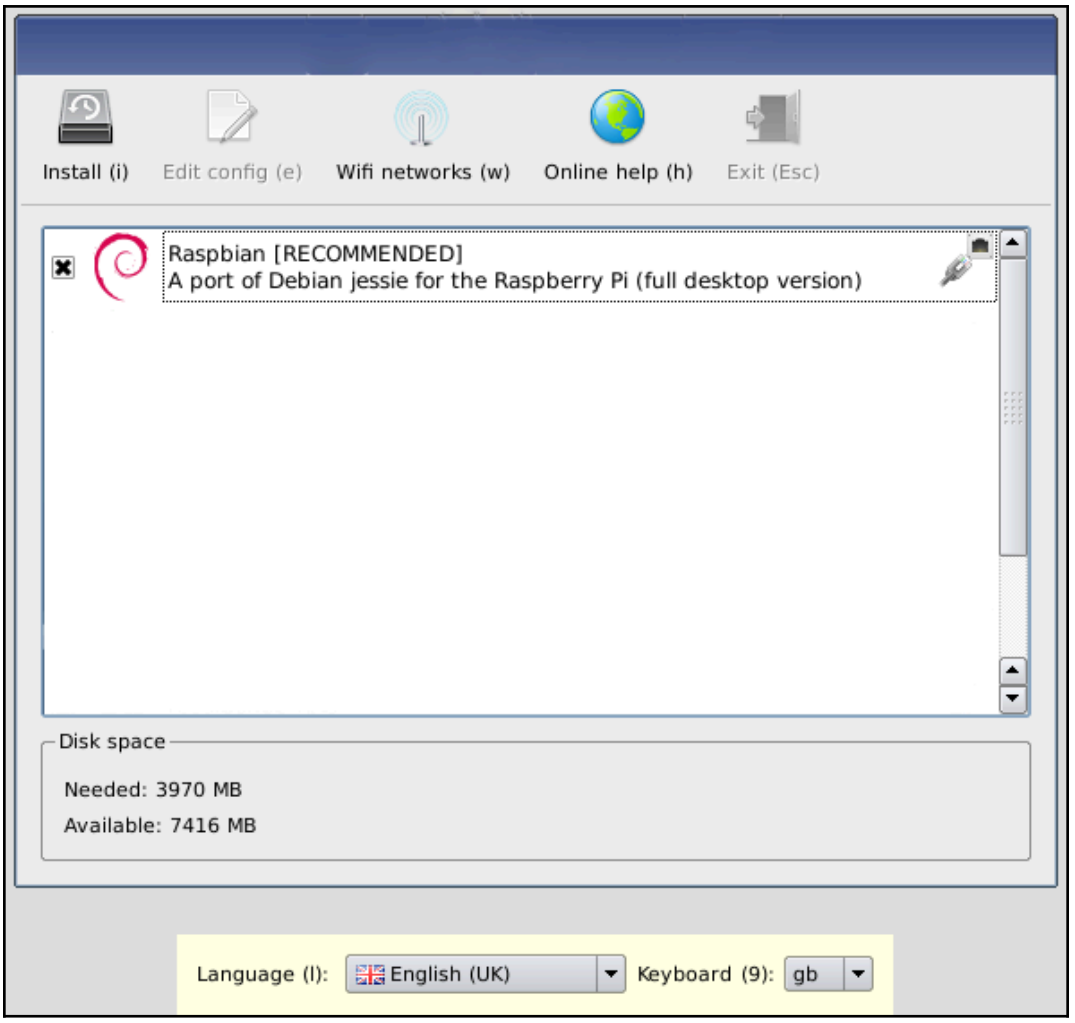
Inserting the microSD card into a USB adapter

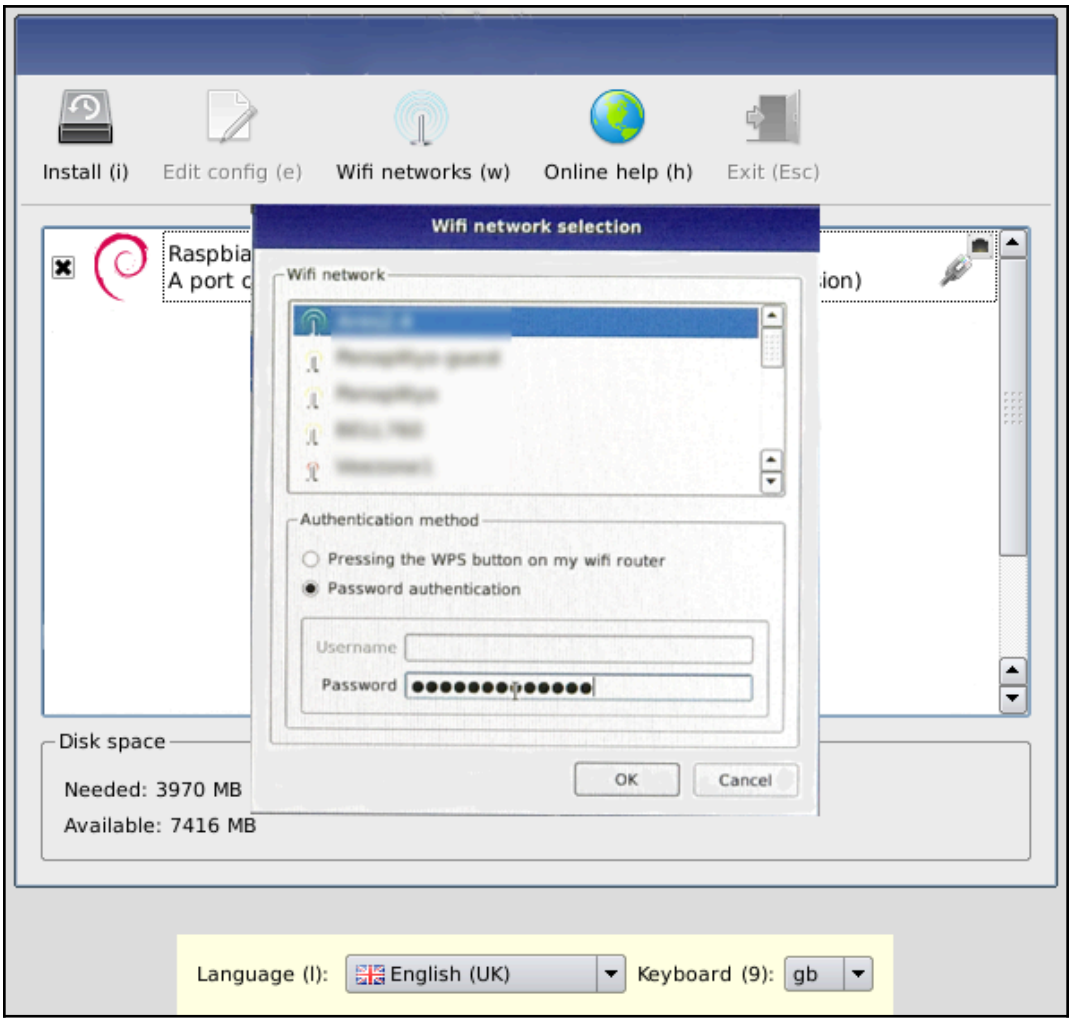


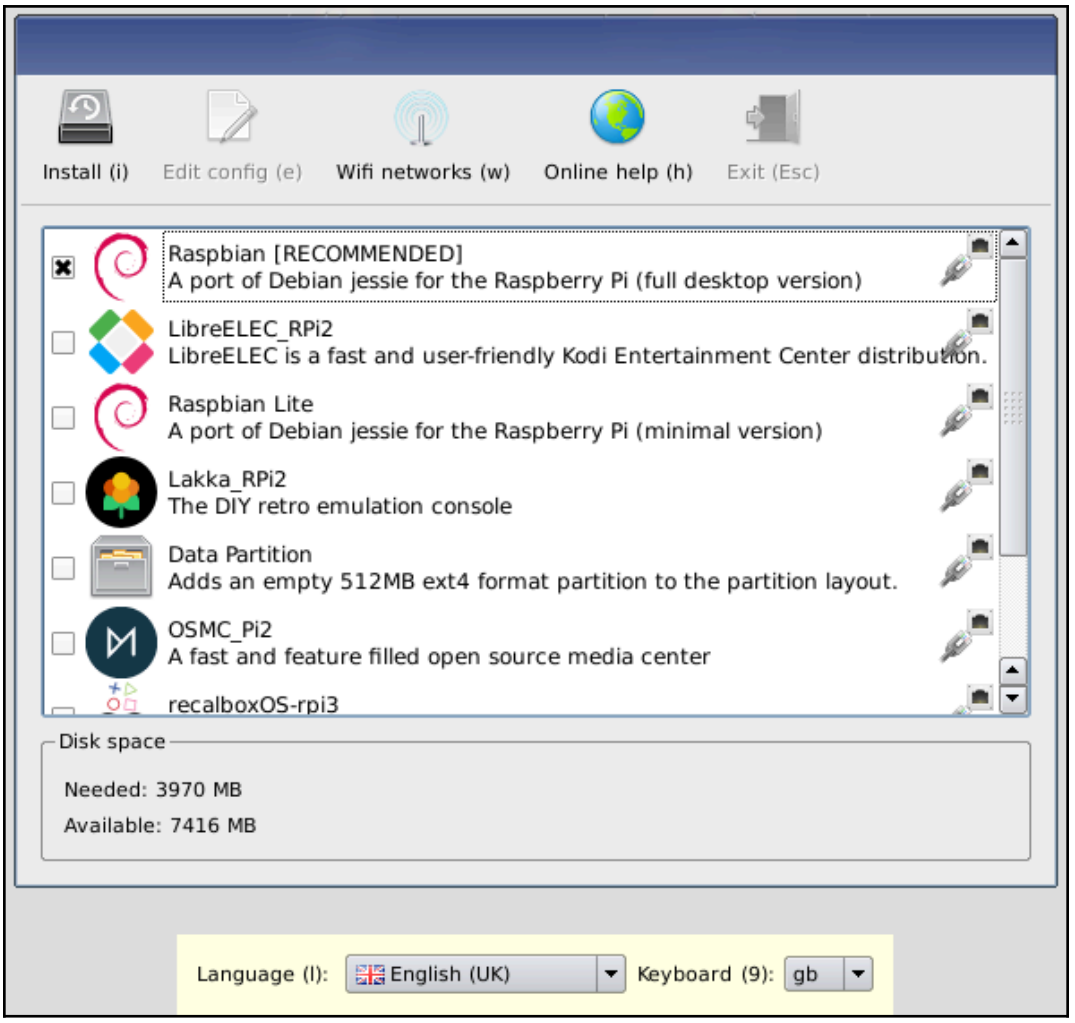
NOOBS File Structure

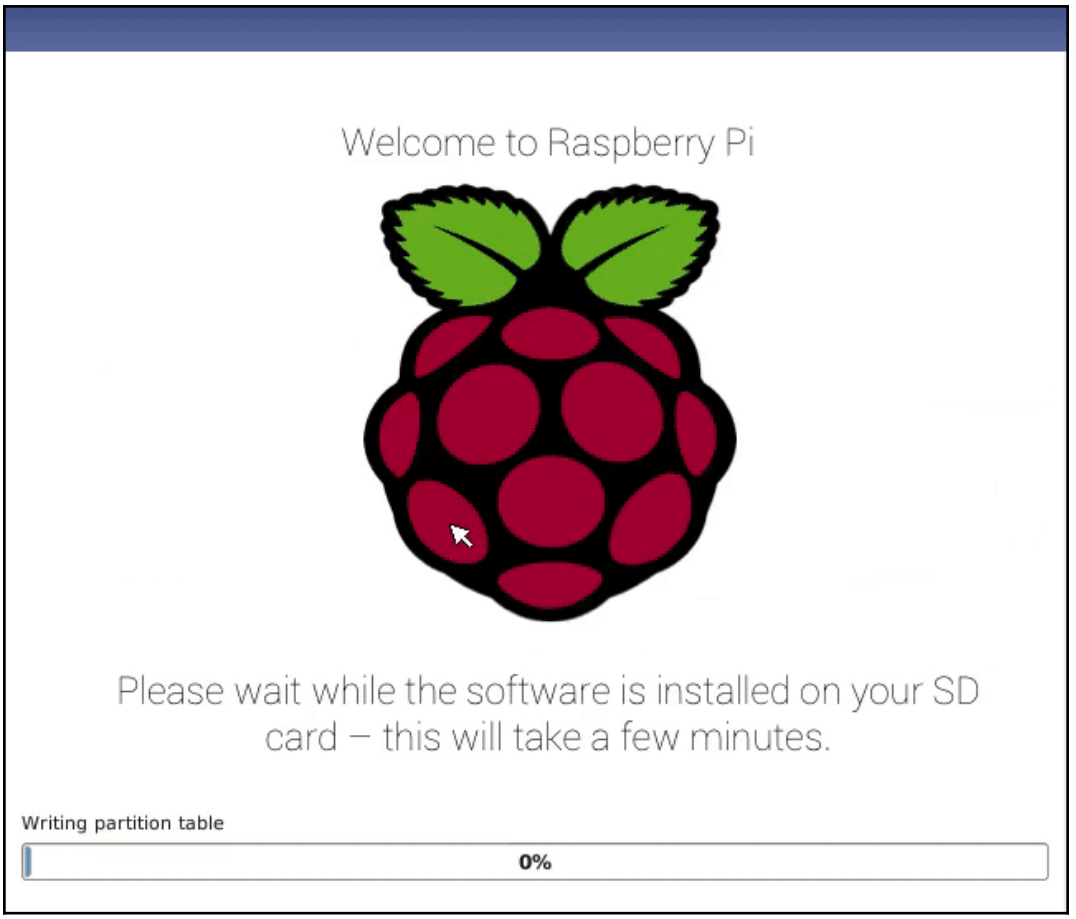


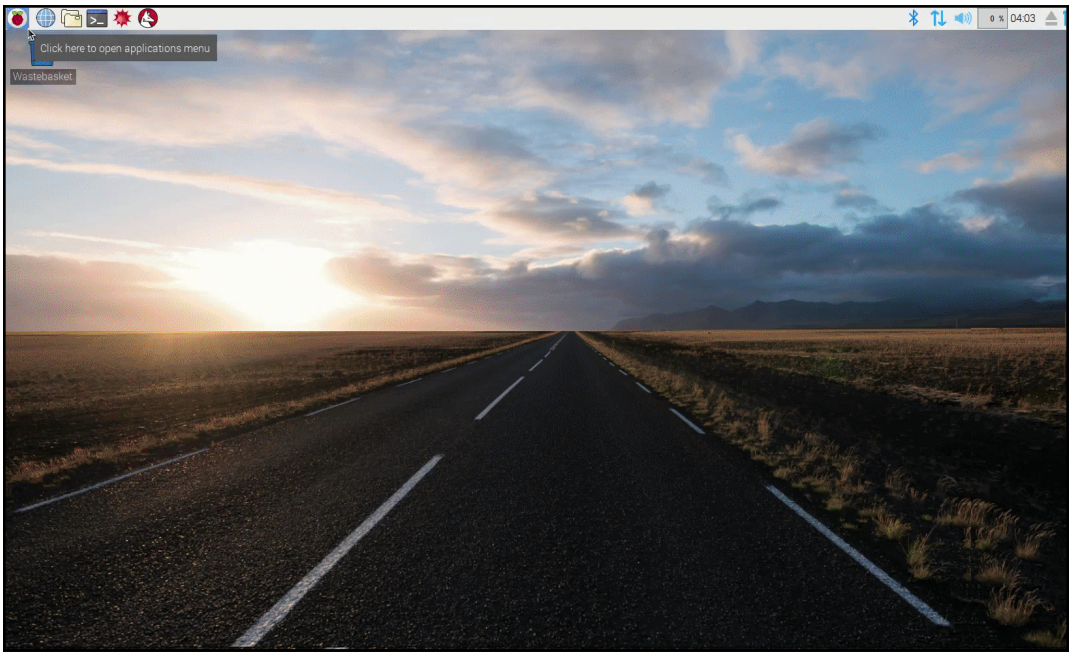
Inserting the microSD into the Raspberry Pi

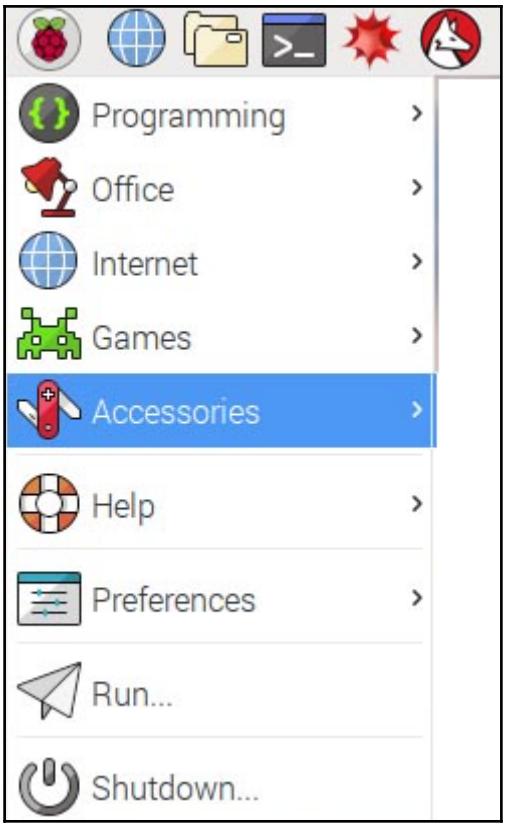

Install (i)
Edit config (e)
Wifi networks (w)
Online help (h)
Exit (Esc)
Raspbian [RECOMMENDED]
A port of Debian jessie for the Raspberry Pi (full desktop version)
Disk space
Needed: 3970 MB
Available: 7416 MB
Language (l):
English (UK)
Keyboard (9):
gb

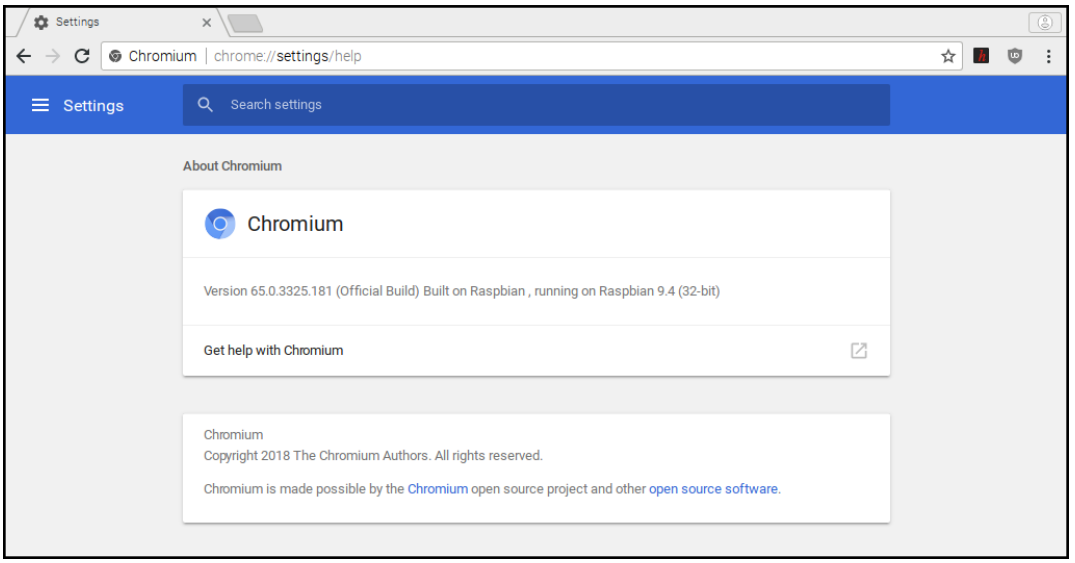

Install (i)
Edit config (e)
Wifi networks (w)
Online help (h)
Exit (Esc)
Wifi network selection
Wifi network
Authentication method
Pressing the WPS button on my wifi router
Password authentication
Username
Password
OK
Cancel
Disk space
Needed: 3970 MB
Available: 7416 MB
Language (l):
English (UK)
Keyboard (9):
gb

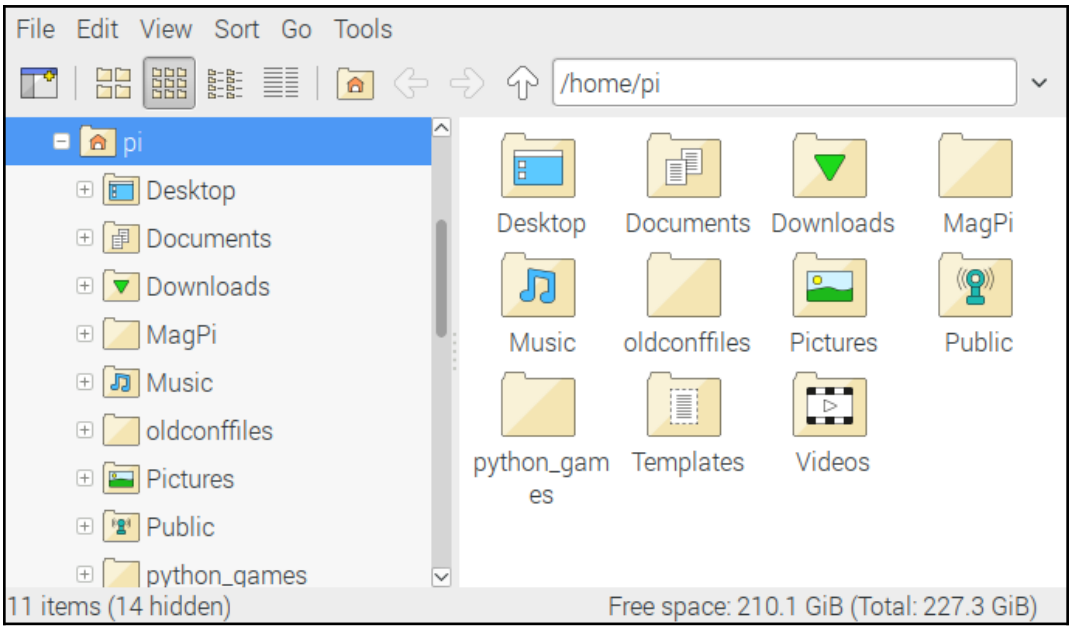

Install (i)
Edit config (e)
Wifi networks (w)
Online help (h)
Exit (Esc)
Raspbian [RECOMMENDED]
A port of Debian jessie for the Raspberry Pi (full desktop version)
LibreELEC_RPi2
LibreELEC is a fast and user-friendly Kodi Entertainment Center distribution.
Raspbian Lite
A port of Debian jessie for the Raspberry Pi (minimal version)
Lakka_RPi2
The DIY retro emulation console
Data Partition
Adds an empty 512MB ext4 format partition to the partition layout.
OSMC_Pi2
A fast and feature filled open source media center
recalboxOS-rpi3
Disk space
Needed: 3970 MB
Available: 7416 MB
Language (l):
English (UK)
Keyboard (9):
gb

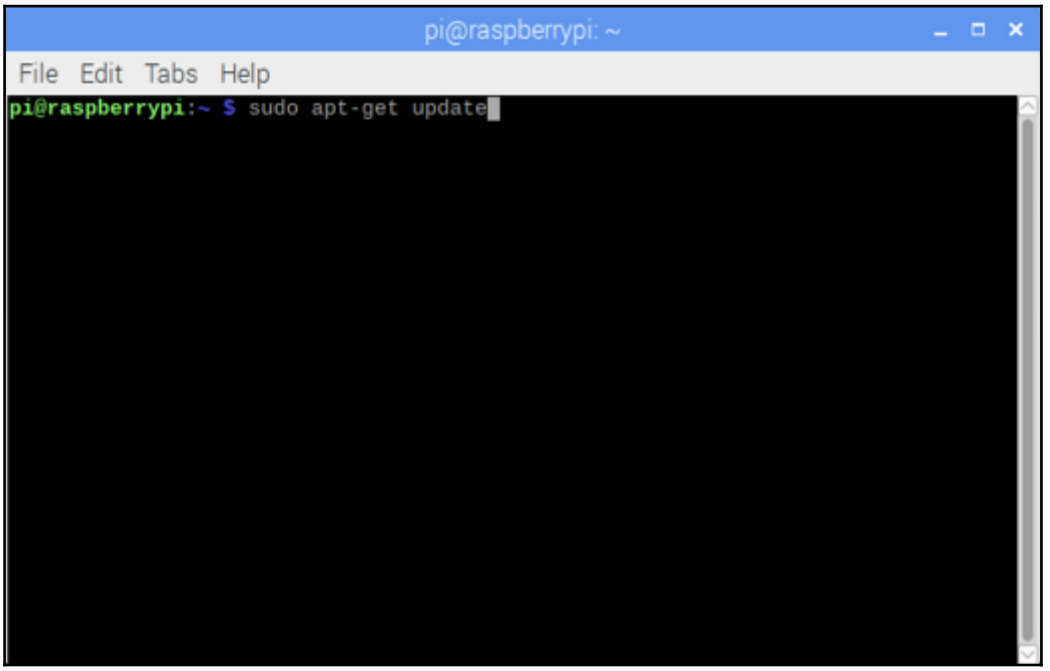

Welcome to Raspberry Pi
Please wait while the software is installed on your SD card – this will take a few minutes.
Writing partition table
0%

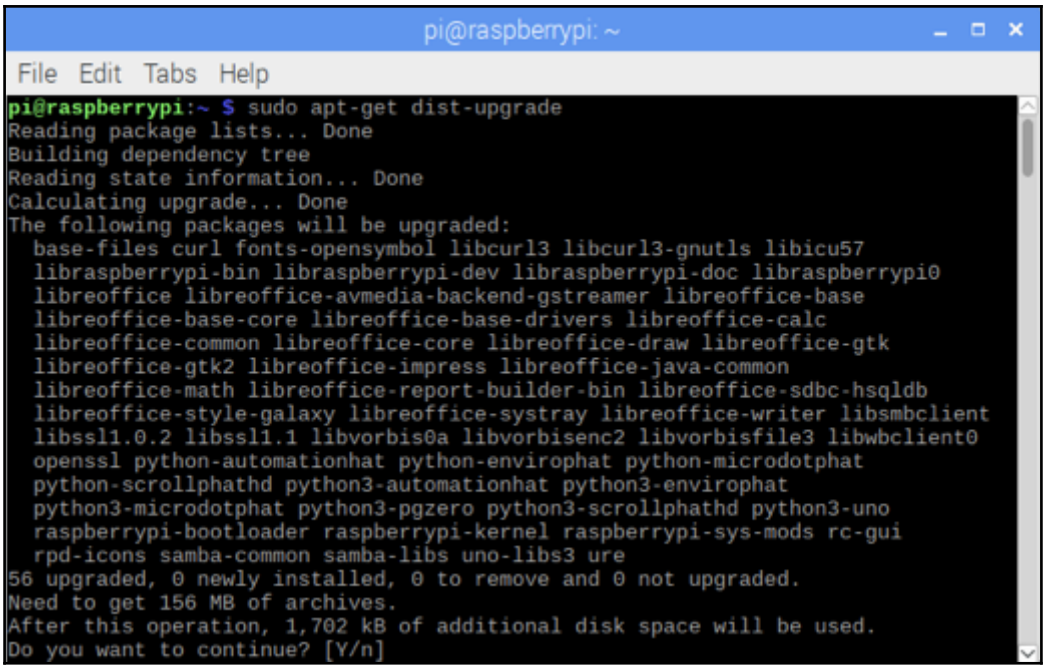

Click here to open applications menu
Wastebasket
0 %
04:03

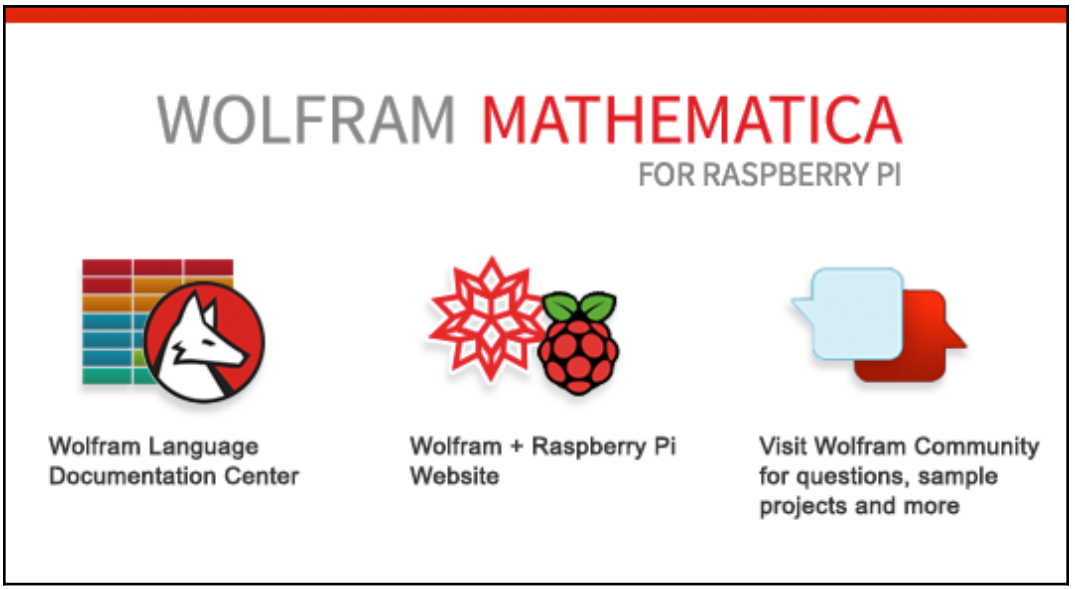

Programming
Office
Internet
Games
Accessories
Help
Preferences
Run...
Shutdown...


Settings
Chromium
chrome://settings/help
Settings
Search settings
About Chromium
Chromium
Version 65.0.3325.181 (Official Build) Built on Raspbian , running on Raspbian 9.4 (32-bit)
Get help with Chromium
Chromium
Copyright 2018 The Chromium Authors. All rights reserved.
Chromium is made possible by the Chromium open source project and other open source software.

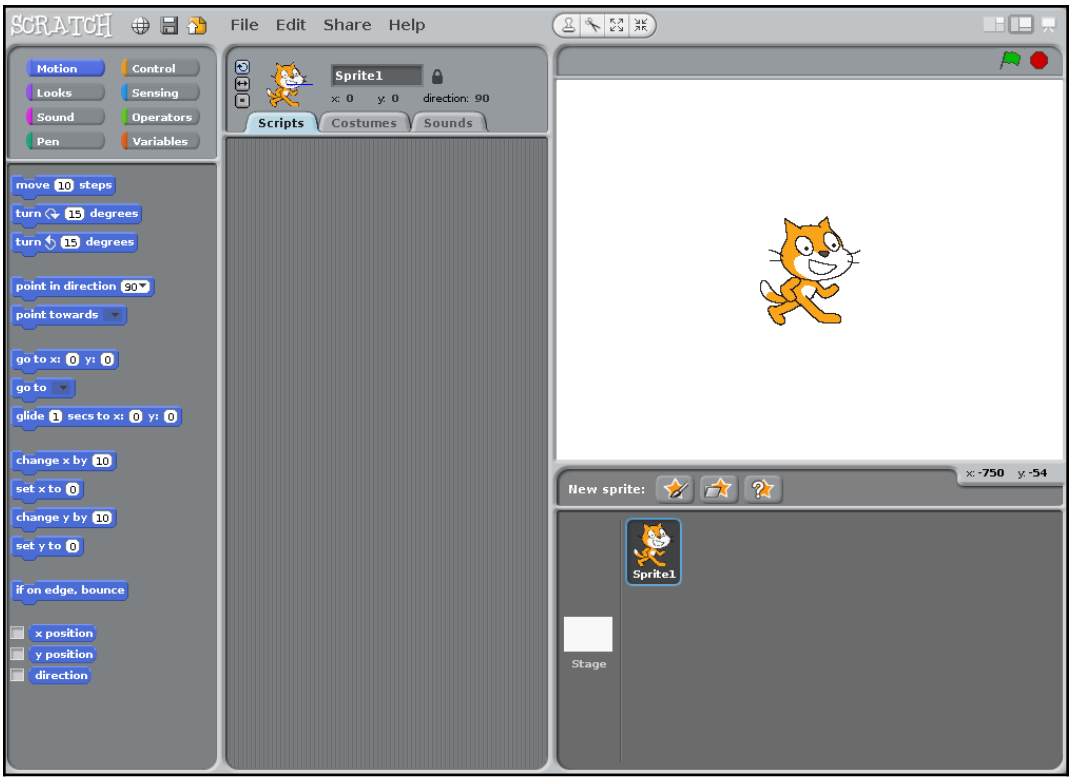

File Edit View Sort Go Tools
/home/pi
pi
Desktop
Documents
Downloads
MagPi
Music
oldconffiles
Pictures
Public
python_games
Templates
Videos
11 items (14 hidden)
Free space: 210.1 GiB (Total: 227.3 GiB)

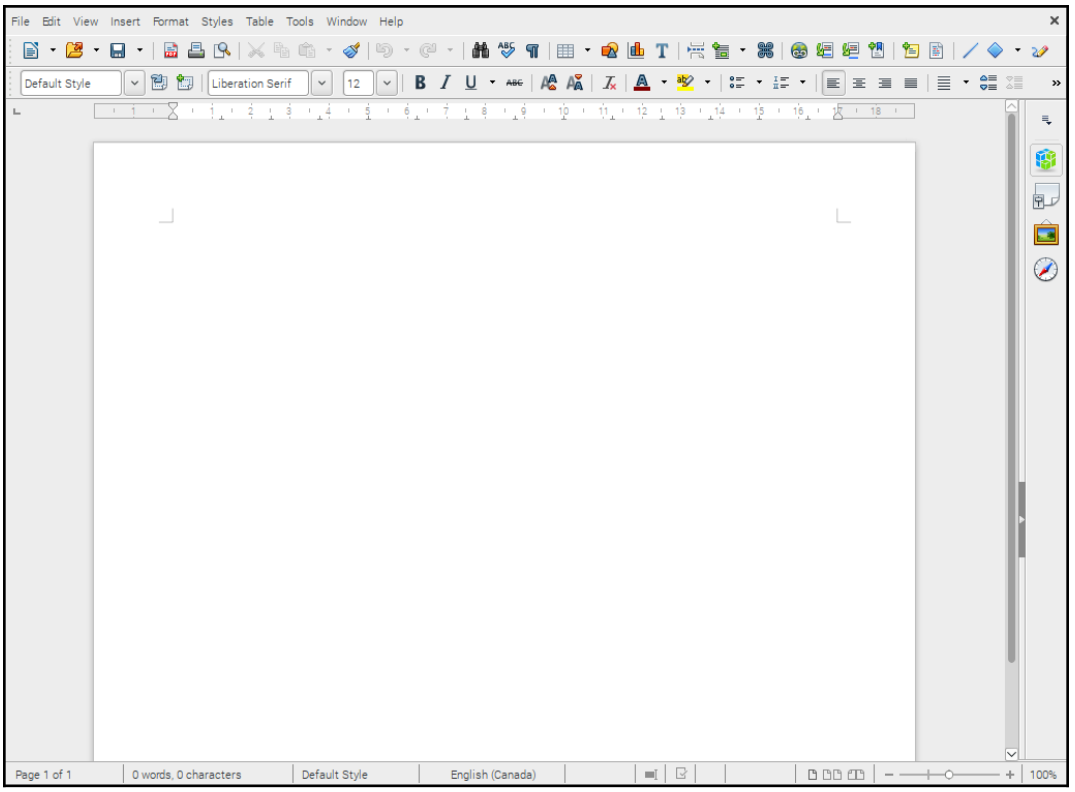

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get update


pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo apt-get dist-upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages will be upgraded:
base-files curl fonts-opensymbol libcurl3 libcurl3-gnutls libicu57
libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0
libreoffice libreoffice-avmedia-backend-gstreamer libreoffice-base
libreoffice-base-core libreoffice-base-drivers libreoffice-calc
libreoffice-common libreoffice-core libreoffice-draw libreoffice-gtk
libreoffice-gtk2 libreoffice-impress libreoffice-java-common
libreoffice-math libreoffice-report-builder-bin libreoffice-sdbc-hsqldb
libreoffice-style-galaxy libreoffice-systray libreoffice-writer libsmbclient
libssl1.0.2 libssl1.1 libvorbis0a libvorbisenc2 libvorbisfile3 libwbclient0
openssl python-automationhat python-envirophat python-microdotphat
python-scrollphathd python3-automationhat python3-envirophat
python3-microdotphat python3-pgzero python3-scrollphathd python3-uno
raspberrypi-bootloader raspberrypi-kernel raspberrypi-sys-mods rc-gui
rpd-icons samba-common samba-libs uno-libs3 ure
56 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 156 MB of archives.
After this operation, 1,702 kB of additional disk space will be used.
Do you want to continue? [Y/n]

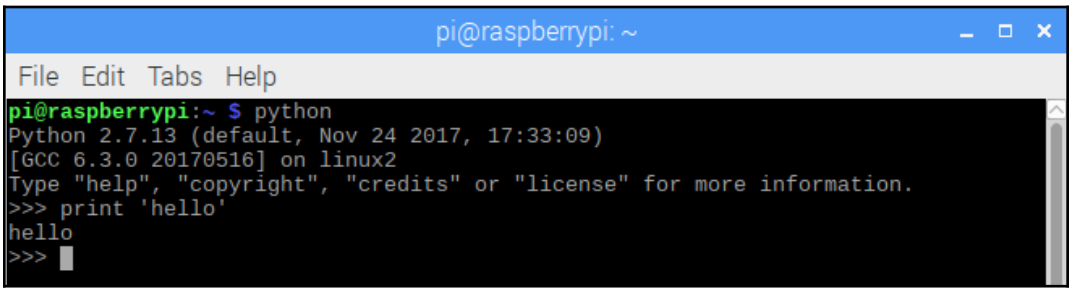

WOLFRAM MATHEMATICA
FOR RASPBERRY PI
Wolfram Language Documentation Center
Wolfram + Raspberry Pi Website
Visit Wolfram Community for questions, sample projects and more

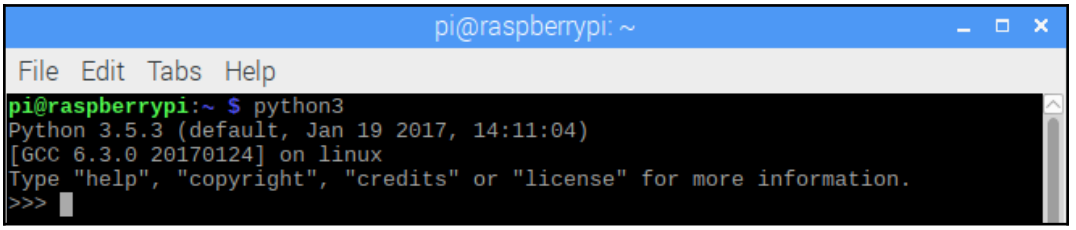

Run
Stop
Rec
Save
Load
Size
Size
Scope
Welcome to
Log
=> Initialised SuperCollider Audio
=> Initialised MIDI subsystems
=> Hello, it's lovely to see
Cues
Buffer 0
Buffer 1
Buffer 2
Buffer 3
Buffer 4
Buffer 5
Help
1 Welcome to Sonic Pi
1.1 Live Coding
1.2 Exploring the Interface
1.3 Learning through Play
2 Synths
2.1 Your First Beeps
Tutorial
Examples
Synths
Fx
Samples
Lang
Sonic Pi v3.0.1 on Raspberry Pi

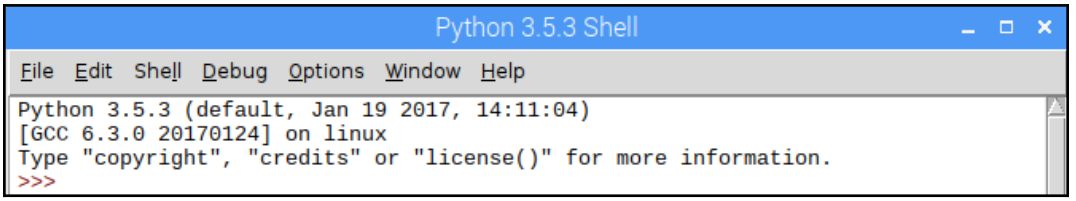

SCRATCH
File Edit Share Help
Motion
Control
Looks
Sensing
Sound
Operators
Pen
Variables
move 10 steps
turn 15 degrees
turn 15 degrees
point in direction 90
point towards
go to x: 0 y: 0
go to
glide 1 secs to x: 0 y: 0
change x by 10
set x to 0
change y by 10
set y to 0
if on edge, bounce
x position
y position
direction
Sprite1
x: 0 y: 0 direction: 90
Scripts
Costumes
Sounds
New sprite:
x: -750 y: -54
Sprite1
Stage

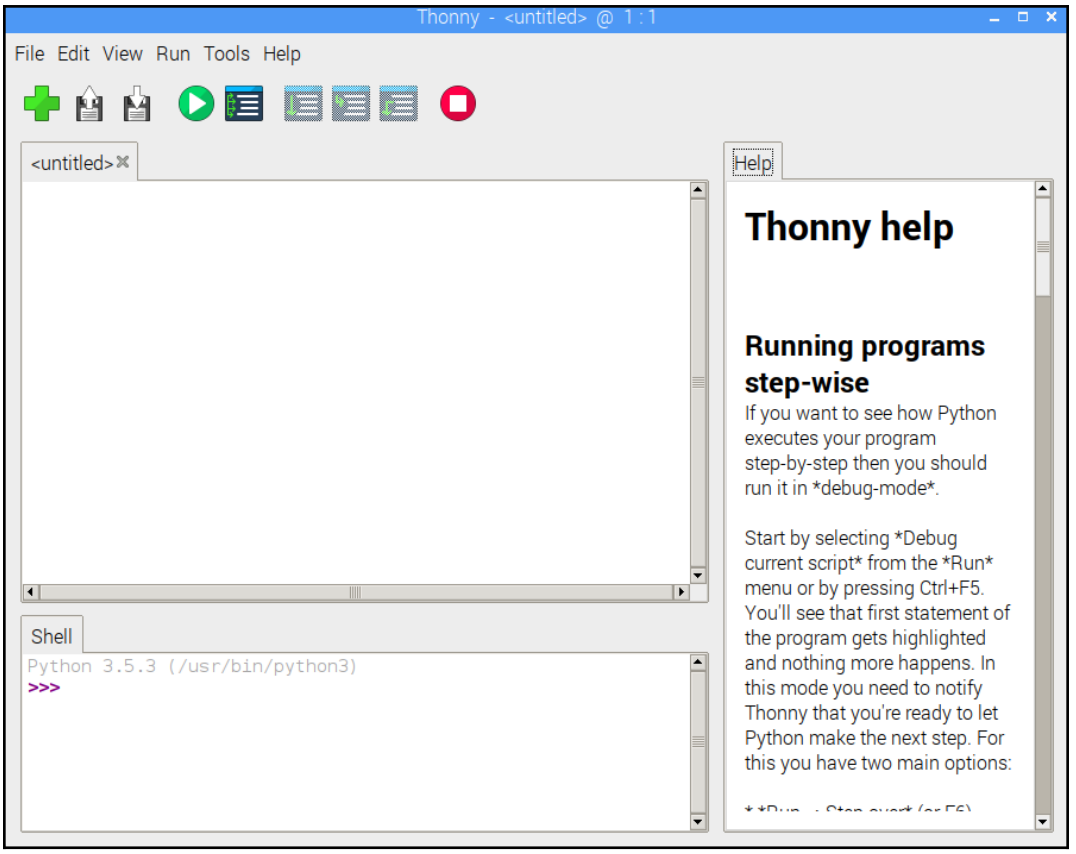

File Edit View Insert Format Styles Table Tools Window Help
Default Style
Liberation Serif
12
Page 1 of 1
0 words, 0 characters
Default Style
English (Canada)
100%

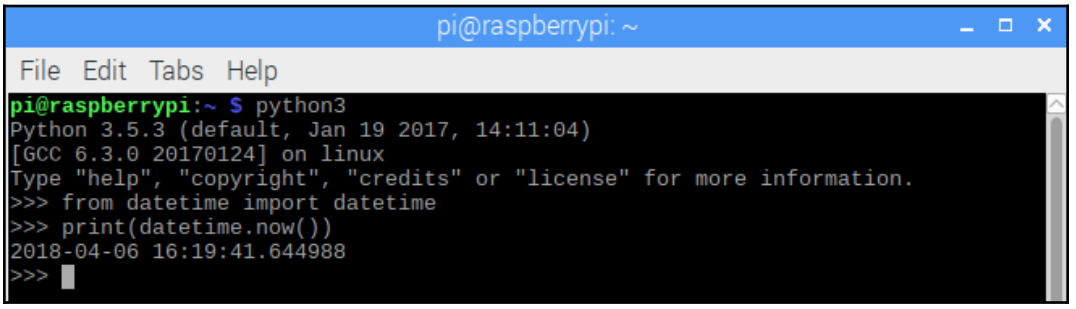
Chapter 11: Writing Python Programs Using Raspberry Pi

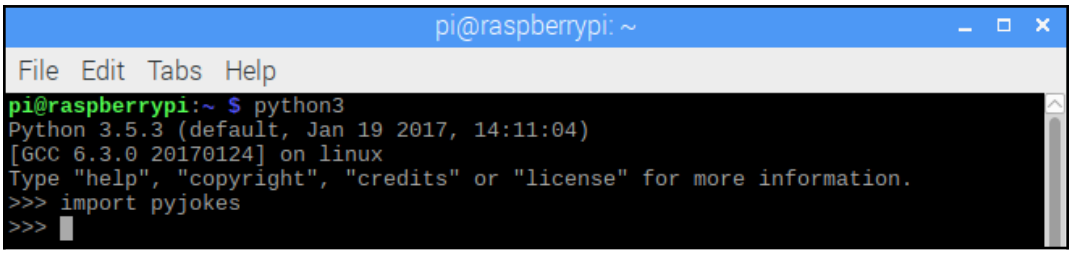


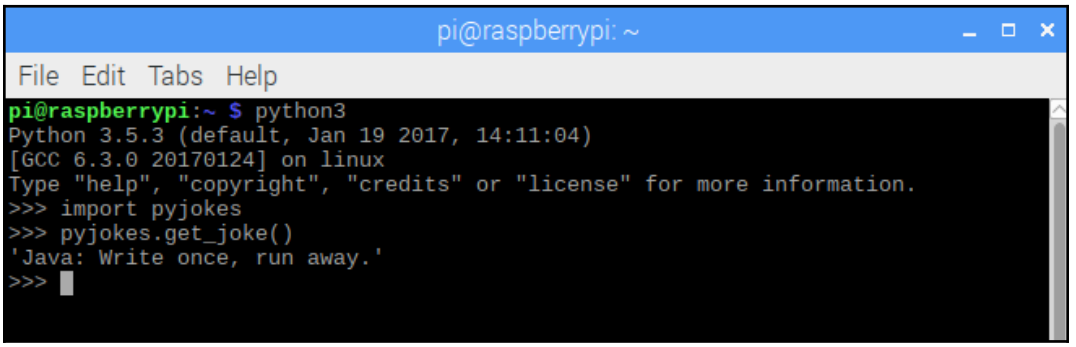


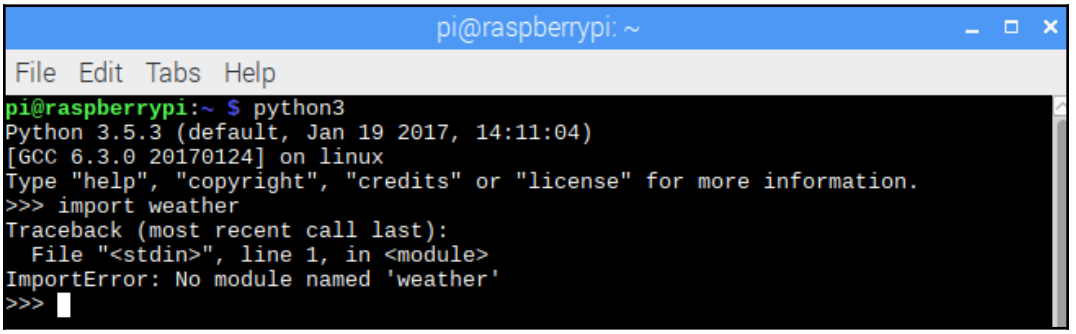


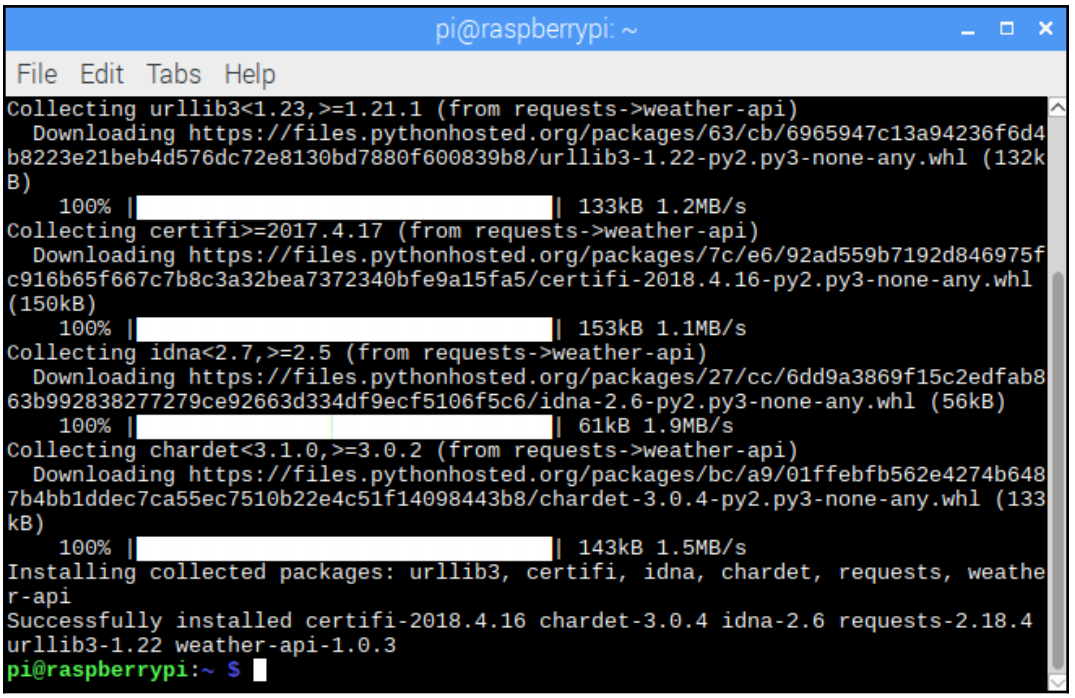

Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
<untitled>
Shell
Python 3.5.3 (/usr/bin/python3)
>>>
Help
Thonny help
Running programs step-wise
If you want to see how Python executes your program step-by-step then you should run it in *debug-mode*.
Start by selecting *Debug current script* from the *Run* menu or by pressing Ctrl+F5. You'll see that first statement of the program gets highlighted and nothing more happens. In this mode you need to notify Thonny that you're ready to let Python make the next step. For this you have two main options:

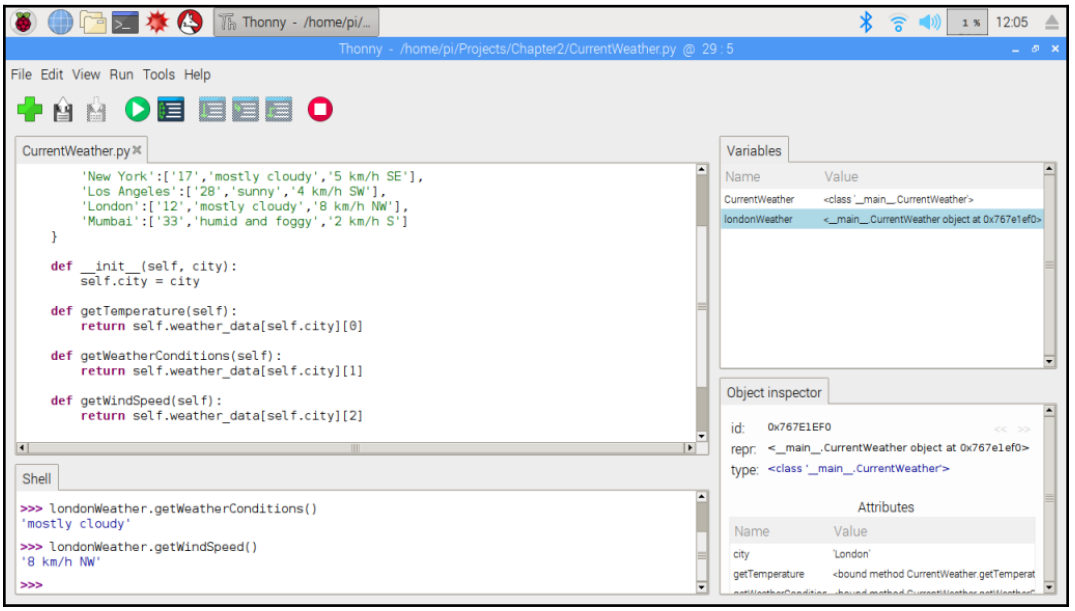

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> from datetime import datetime
>>> print(datetime.now())
2018-04-06 16:19:41.644988
>>>


pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyjokes
>>>

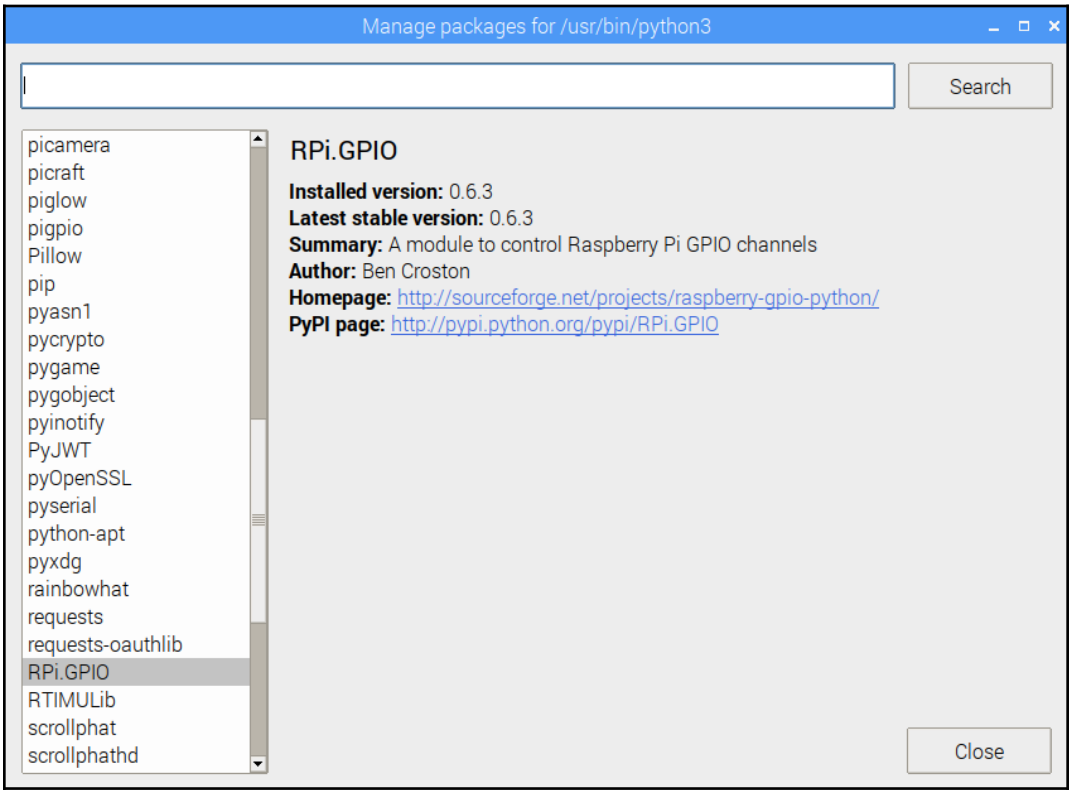

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pyjokes
>>> pyjokes.get_joke()
'Java: Write once, run away.'
>>>


pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ python3
Python 3.5.3 (default, Jan 19 2017, 14:11:04)
[GCC 6.3.0 20170124] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import weather
Traceback (most recent call last):
File "<stdin>", line 1, in <module>
ImportError: No module named 'weather'
>>>

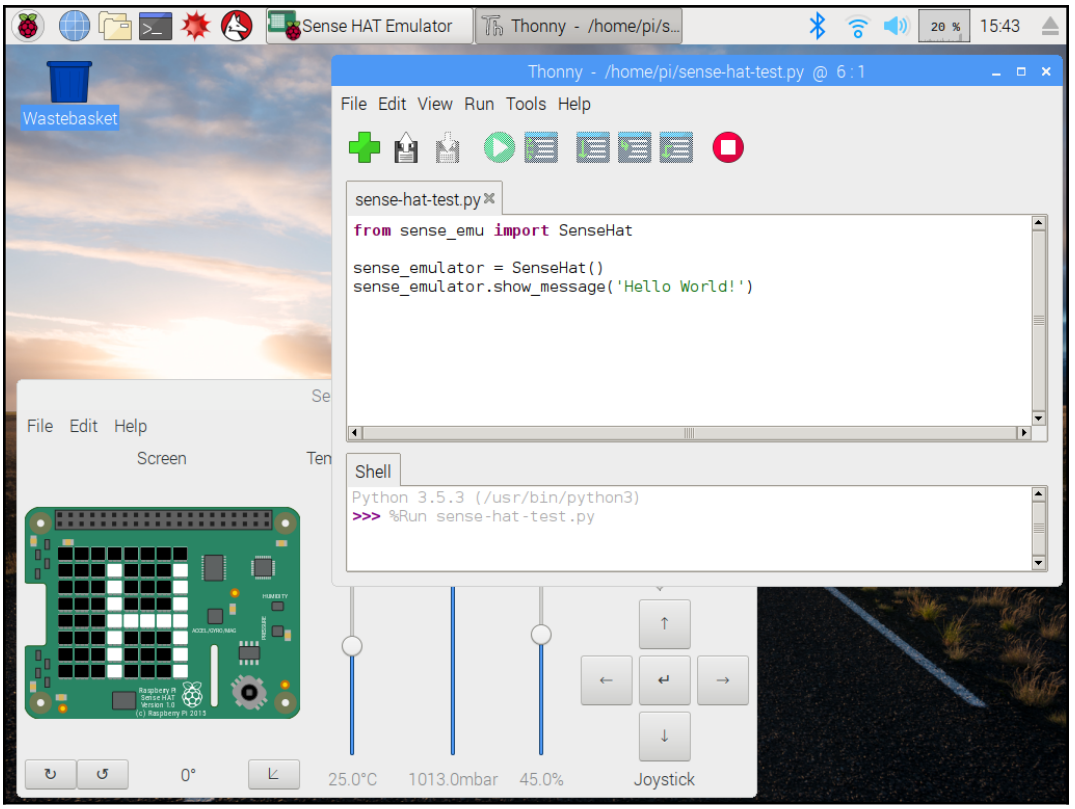

pi@raspberrypi: ~
File Edit Tabs Help
Collecting urllib3<1.23,>=1.21.1 (from requests->weather-api)
Downloading https://files.pythonhosted.org/packages/63/cb/6965947c13a94236f6d4b8223e21beb4d576dc72e8130bd7880f600839b8/urllib3-1.22-py2.py3-none-any.whl (132kB)
100% | 133kB 1.2MB/s
Collecting certifi>=2017.4.17 (from requests->weather-api)
Downloading https://files.pythonhosted.org/packages/7c/e6/92ad559b7192d846975fc916b65f667c7b8c3a32bea7372340bfe9a15fa5/certifi-2018.4.16-py2.py3-none-any.whl (150kB)
100% | 153kB 1.1MB/s
Collecting idna<2.7,>=2.5 (from requests->weather-api)
Downloading https://files.pythonhosted.org/packages/27/cc/6dd9a3869f15c2edfab863b992838277279ce92663d334df9ecf5106f5c6/idna-2.6-py2.py3-none-any.whl (56kB)
100% | 61kB 1.9MB/s
Collecting chardet<3.1.0,>=3.0.2 (from requests->weather-api)
Downloading https://files.pythonhosted.org/packages/bc/a9/01ffebfb562e4274b6487b4bb1ddec7ca55ec7510b22e4c51f14098443b8/chardet-3.0.4-py2.py3-none-any.whl (133kB)
100% | 143kB 1.5MB/s
Installing collected packages: urllib3, certifi, idna, chardet, requests, weather-api
Successfully installed certifi-2018.4.16 chardet-3.0.4 idna-2.6 requests-2.18.4 urllib3-1.22 weather-api-1.0.3
pi@raspberrypi:~ $

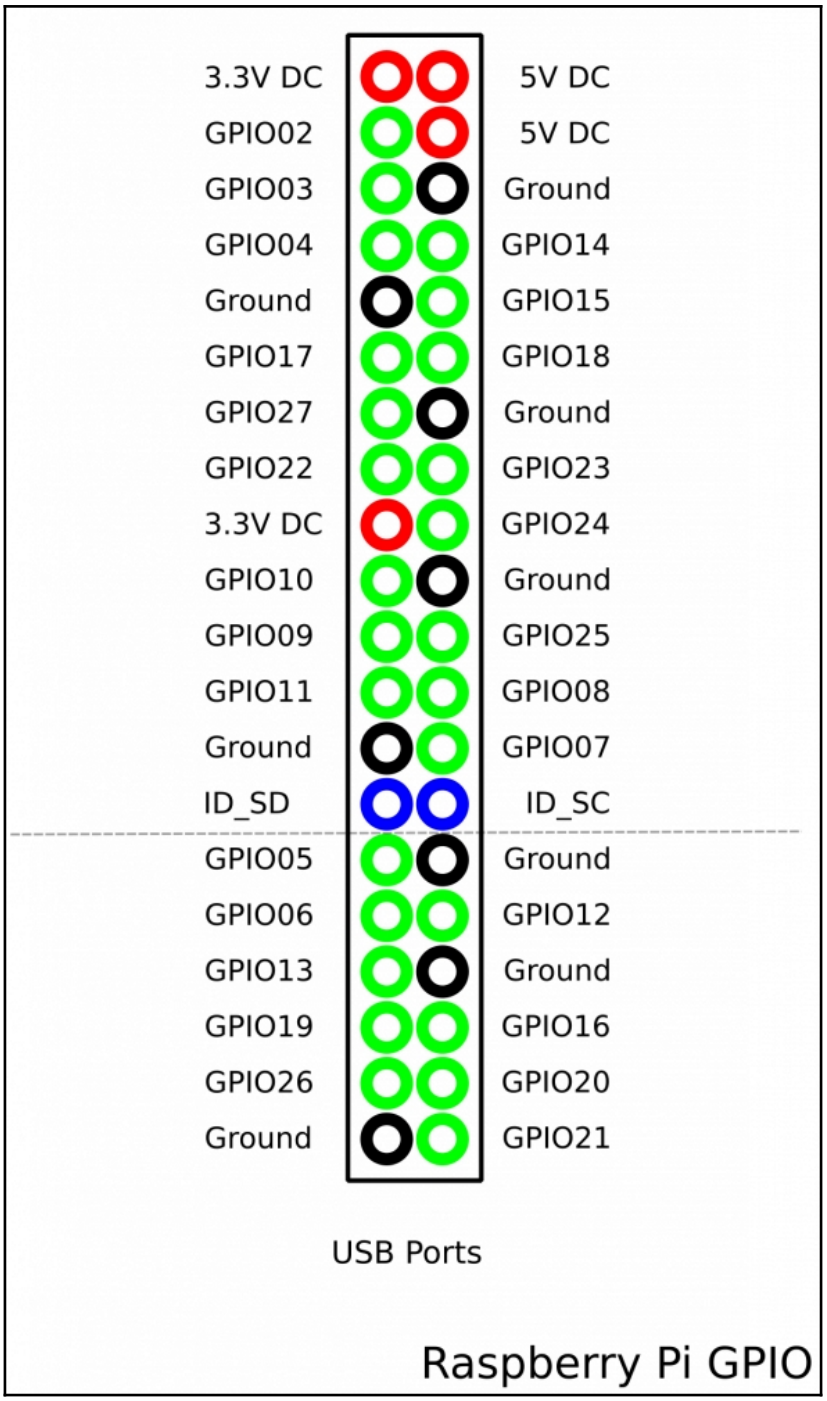

Thonny - /home/pi/...
12:05
Thonny - /home/pi/Projects/Chapter2/CurrentWeather.py @ 29 : 5
File Edit View Run Tools Help
CurrentWeather.py
'New York':['17','mostly cloudy','5 km/h SE'],
'Los Angeles':['28','sunny','4 km/h SW'],
'London':['12','mostly cloudy','8 km/h NW'],
'Mumbai':['33','humid and foggy','2 km/h S']
}
def __init__(self, city):
self.city = city
def getTemperature(self):
return self.weather_data[self.city][0]
def getWeatherConditions(self):
return self.weather_data[self.city][1]
def getWindSpeed(self):
return self.weather_data[self.city][2]
Variables
Name Value
CurrentWeather <class '__main__.CurrentWeather'>
londonWeather <__main__.CurrentWeather object at 0x767e1ef0>
Object inspector
id: 0x767E1EF0
repr: <__main__.CurrentWeather object at 0x767e1ef0>
type: <class '__main__.CurrentWeather'>
Attributes
Name Value
city 'London'
getTemperature <bound method CurrentWeather.getTemperat
Shell
>>> londonWeather.getWeatherConditions()
'mostly cloudy'
>>> londonWeather.getWindSpeed()
'8 km/h NW'
>>>

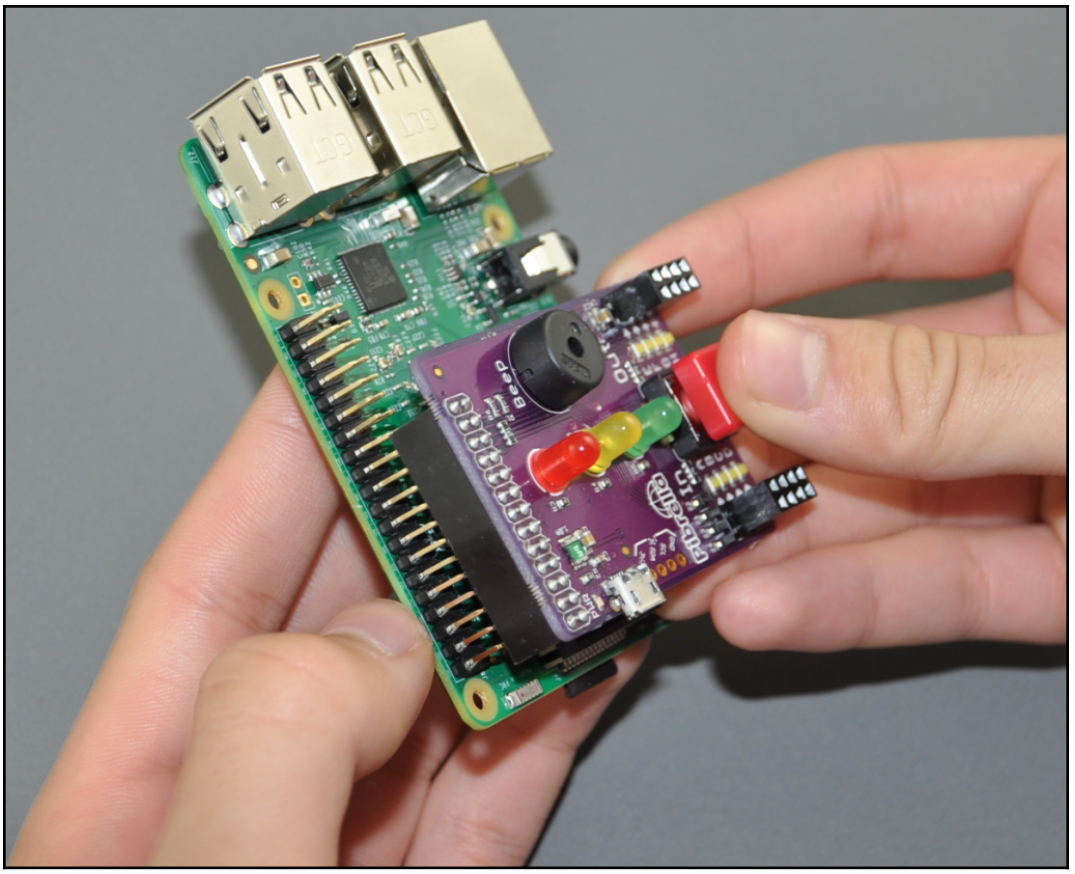
Chapter 12: Using the GPIO to Connect to the Outside World

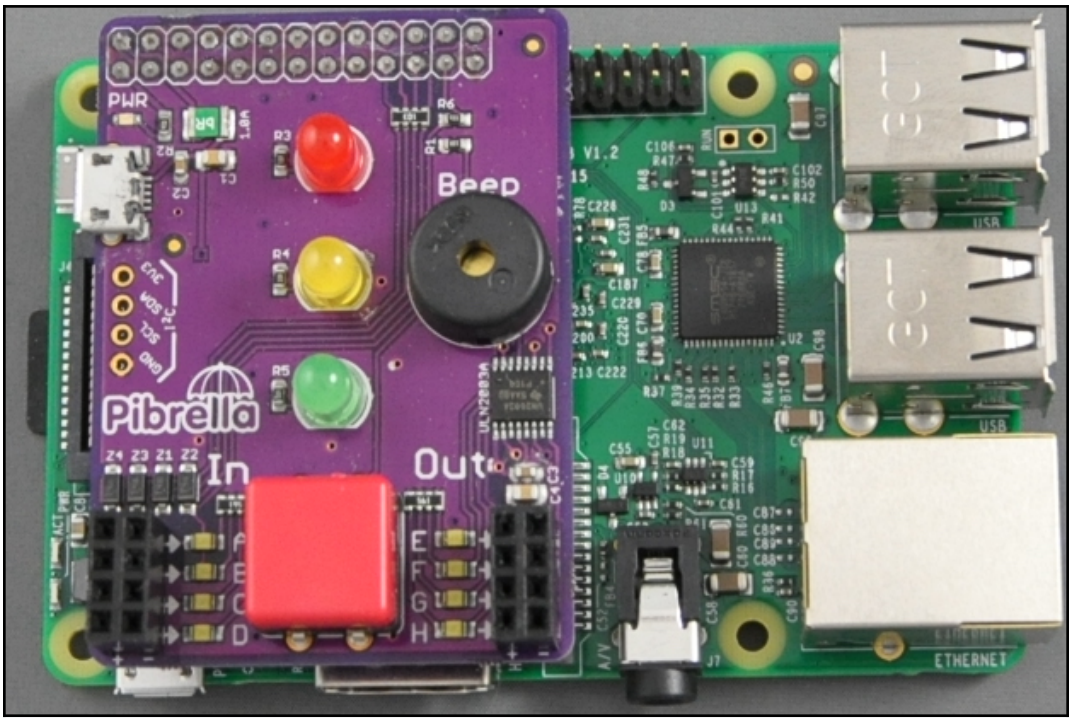


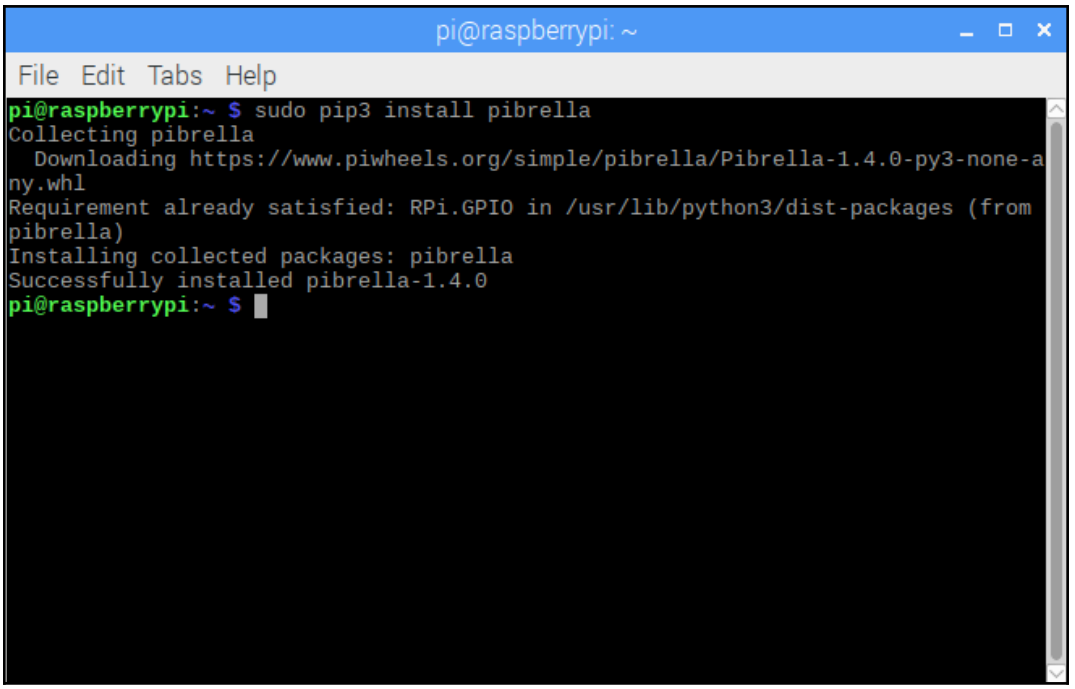

Raspberry Pi 3 Model B V1.2
© Raspberry Pi 2015
BROADCOM
ETHERNET

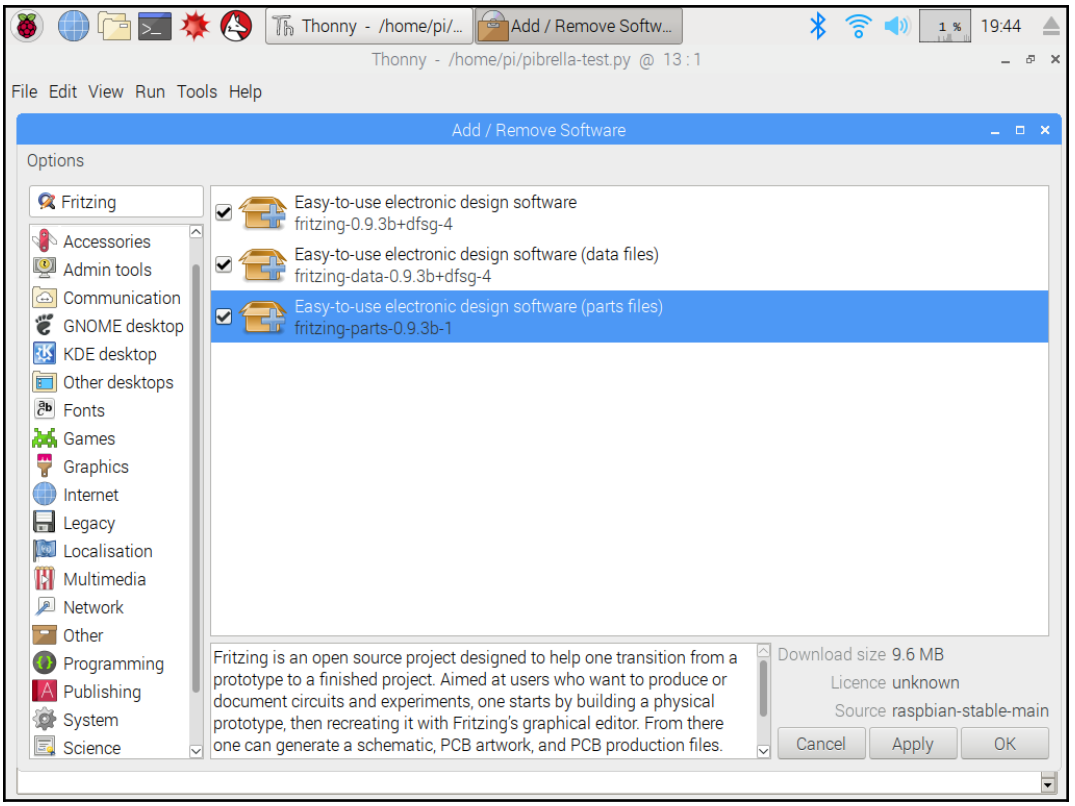

Sense HAT Emulator
Thonny - /home/pi/s...
20 %
15:43
Wastebasket
Thonny - /home/pi/sense-hat-test.py @ 6 : 1
File Edit View Run Tools Help
sense-hat-test.py
from sense_emu import SenseHat
sense_emulator = SenseHat()
sense_emulator.show_message('Hello World!')
Shell
Python 3.5.3 (/usr/bin/python3)
>>> %Run sense-hat-test.py
File Edit Help
Screen
0°
25.0°C
1013.0mbar
45.0%
Joystick

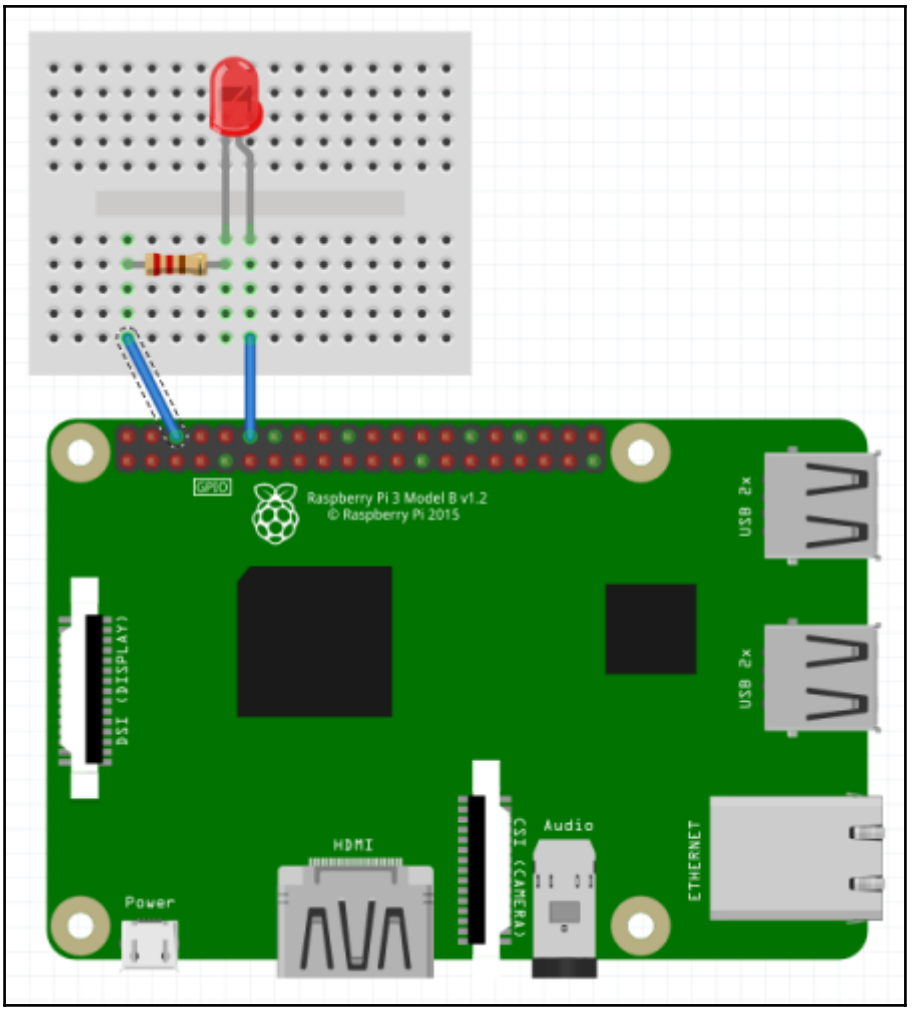

3.3V DC
5V DC
GPIO02
5V DC
GPIO03
Ground
GPIO04
GPIO14
Ground
GPIO15
GPIO17
GPIO18
GPIO27
Ground
GPIO22
GPIO23
3.3V DC
GPIO24
GPIO10
Ground
GPIO09
GPIO25
GPIO11
GPIO08
Ground
GPIO07
ID_SD
ID_SC
GPIO05
Ground
GPIO06
GPIO12
GPIO13
Ground
GPIO19
GPIO16
GPIO26
GPIO20
Ground
GPIO21
USB Ports
Raspberry Pi GPIO

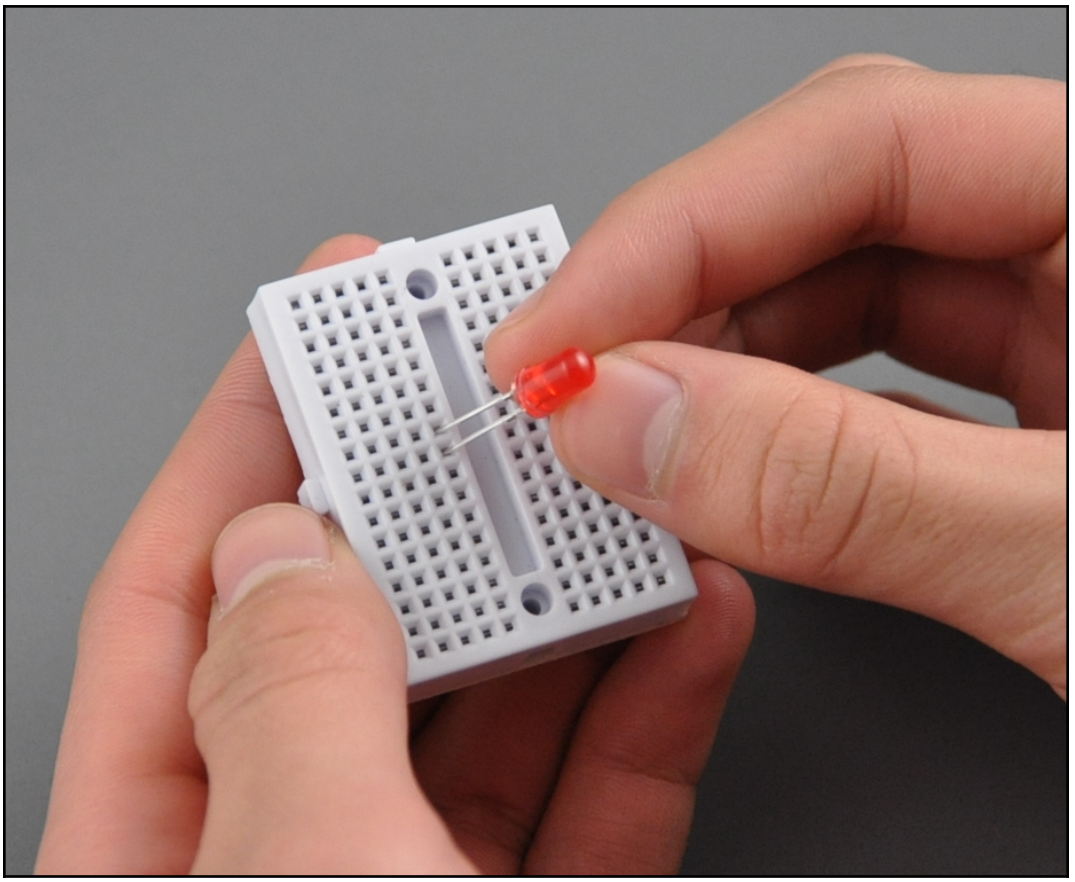


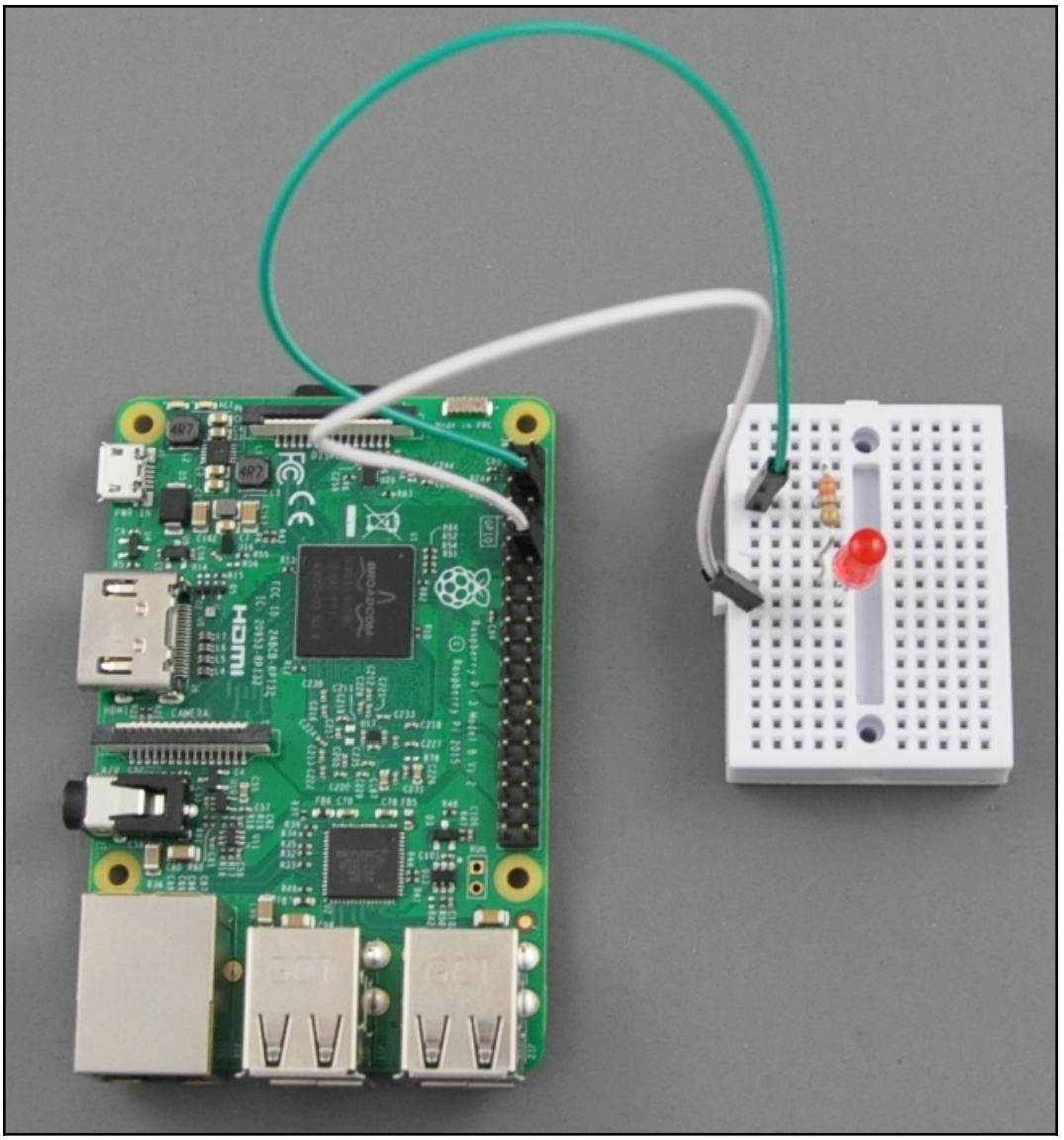

PWR
Beep
Pibrella
In
Out
A
B
C
D
E
F
G
H
ETHERNET
USB


pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo pip3 install pibrella
Collecting pibrella
Downloading https://www.piwheels.org/simple/pibrella/Pibrella-1.4.0-py3-none-any.whl
Requirement already satisfied: RPi.GPIO in /usr/lib/python3/dist-packages (from pibrella)
Installing collected packages: pibrella
Successfully installed pibrella-1.4.0
pi@raspberrypi:~ $

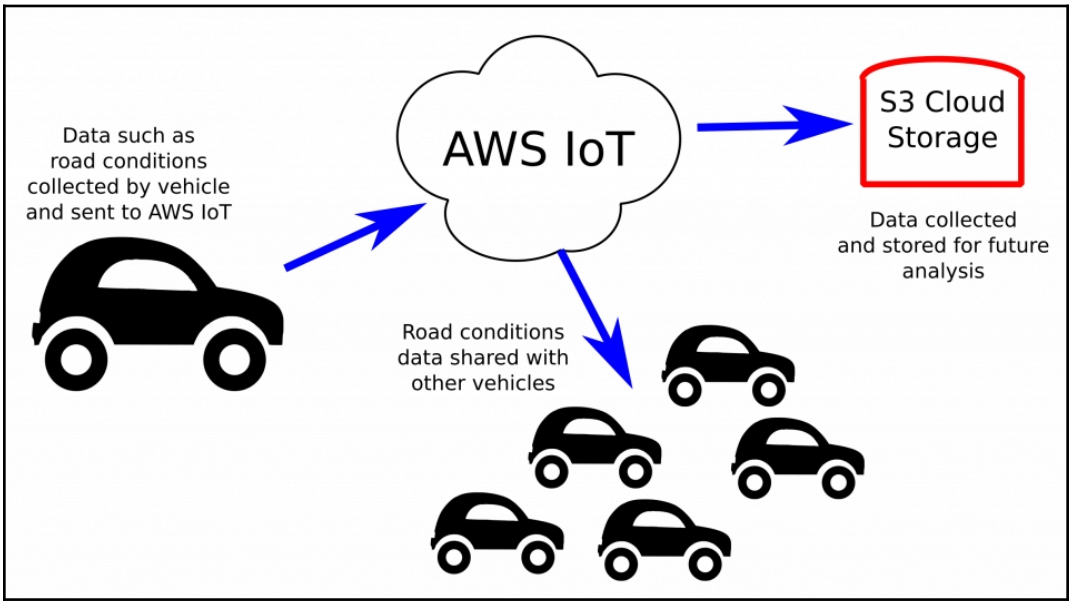

Thonny - /home/pi/...
Add / Remove Softw...
1 %
19:44
Thonny - /home/pi/pibrella-test.py @ 13 : 1
File Edit View Run Tools Help
Add / Remove Software
Options
Fritzing
Accessories
Admin tools
Communication
GNOME desktop
KDE desktop
Other desktops
Fonts
Games
Graphics
Internet
Legacy
Localisation
Multimedia
Network
Other
Programming
Publishing
System
Science
Easy-to-use electronic design software
fritzing-0.9.3b+dfsg-4
Easy-to-use electronic design software (data files)
fritzing-data-0.9.3b+dfsg-4
Easy-to-use electronic design software (parts files)
fritzing-parts-0.9.3b-1
Fritzing is an open source project designed to help one transition from a prototype to a finished project. Aimed at users who want to produce or document circuits and experiments, one starts by building a physical prototype, then recreating it with Fritzing's graphical editor. From there one can generate a schematic, PCB artwork, and PCB production files.
Download size 9.6 MB
Licence unknown
Source raspbian-stable-main
Cancel
Apply
OK

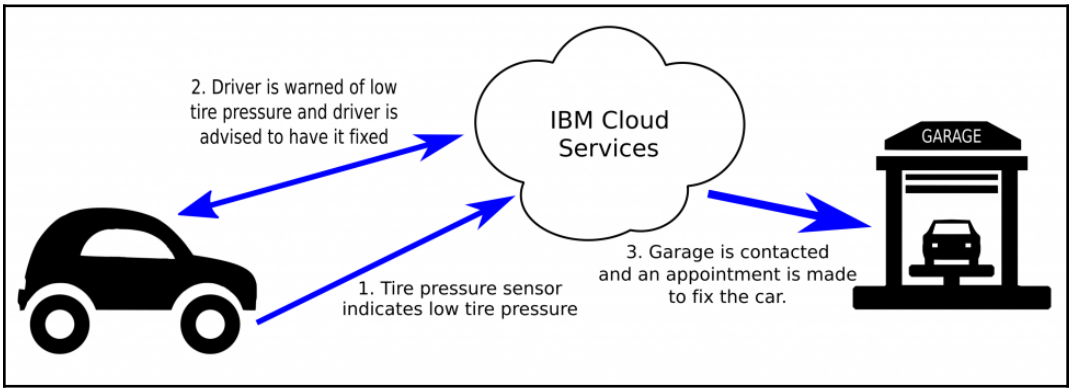

GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power

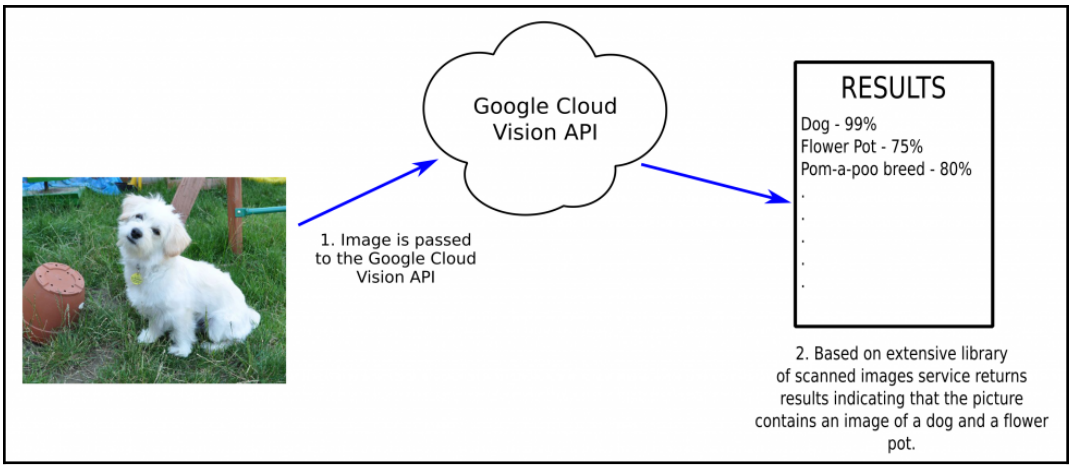


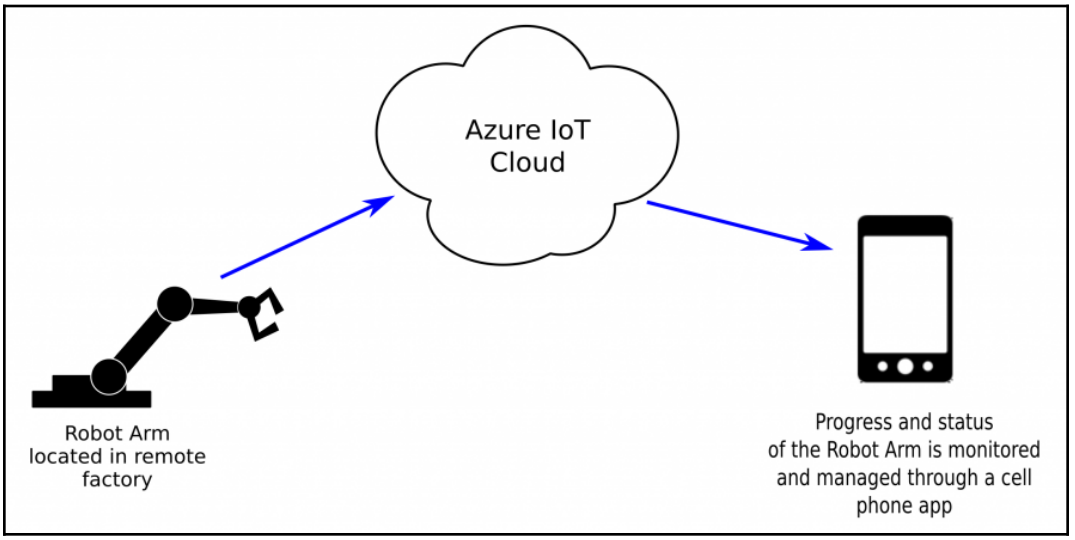


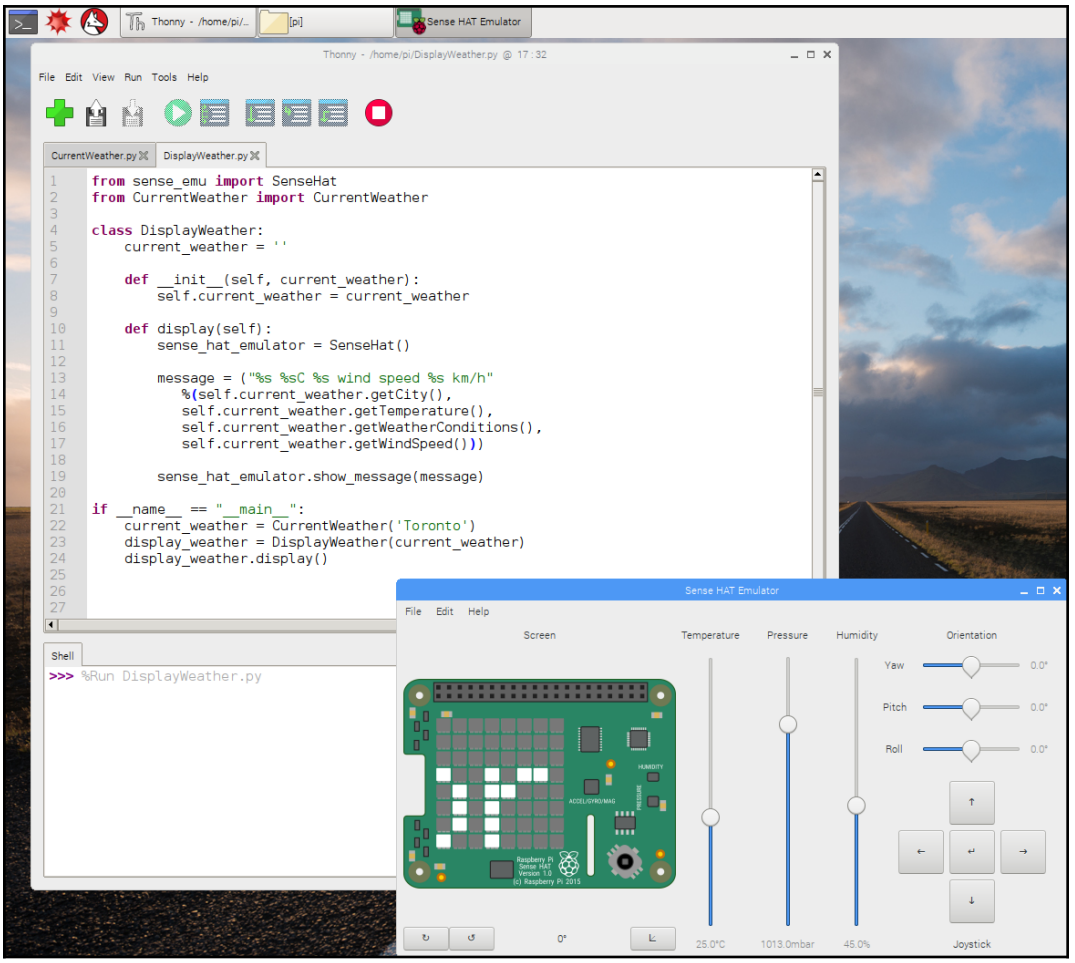
Chapter 13: Subscribing to Web Services



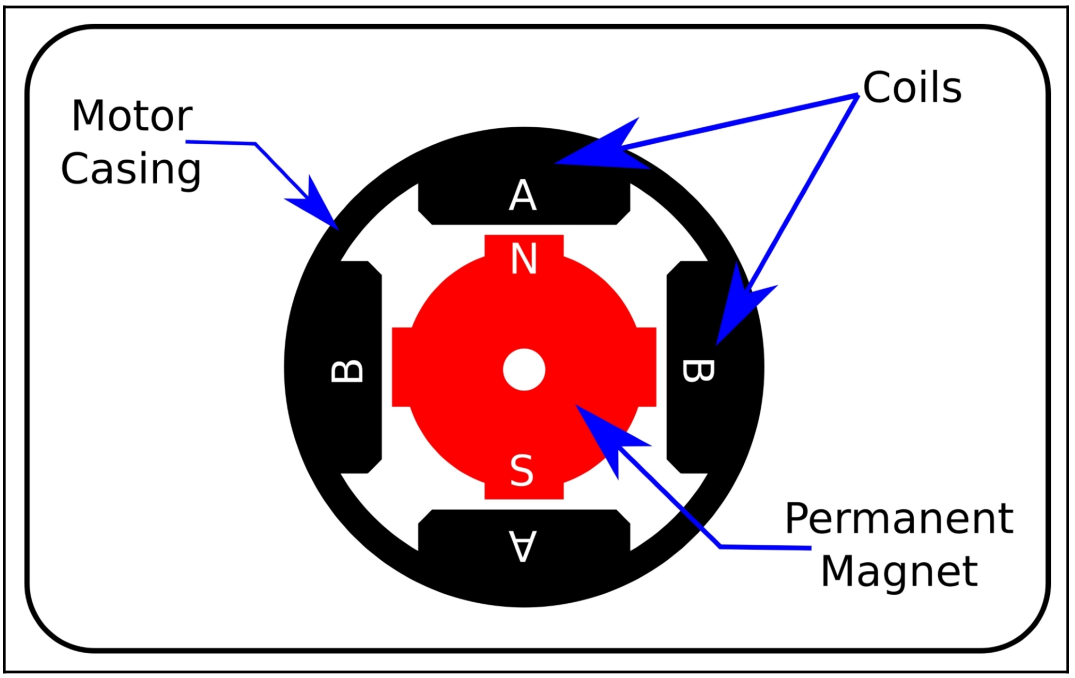


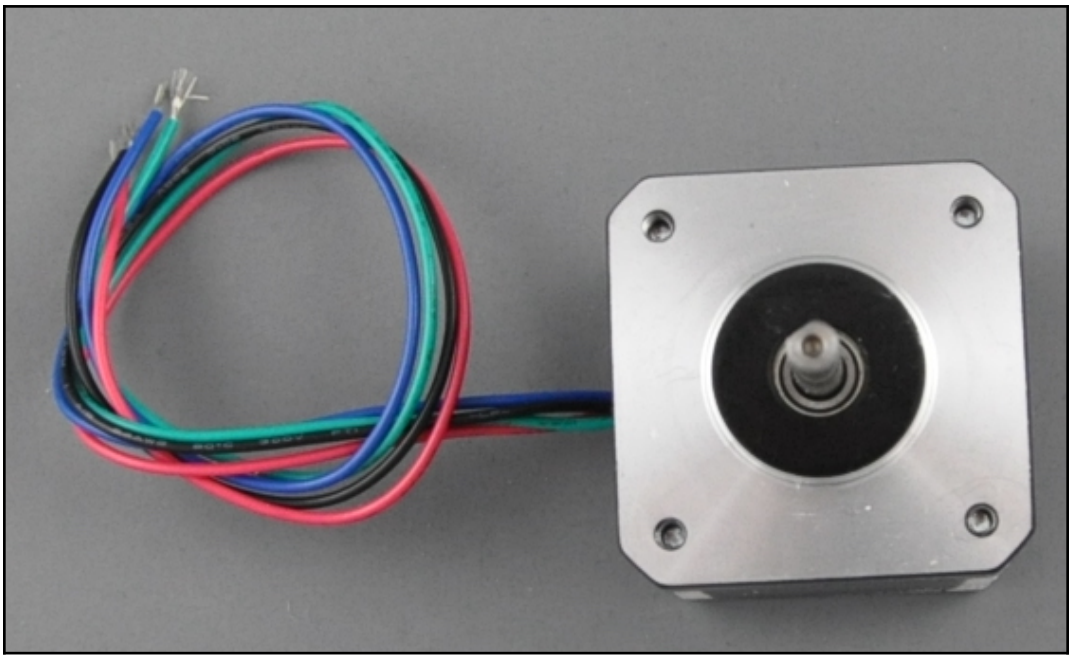

Google Cloud
Vision API
RESULTS
Dog - 99%
Flower Pot - 75%
Pom-a-poo breed - 80%
1. Image is passed
to the Google Cloud
Vision API
2. Based on extensive library
of scanned images service returns
results indicating that the picture
contains an image of a dog and a flower
pot.

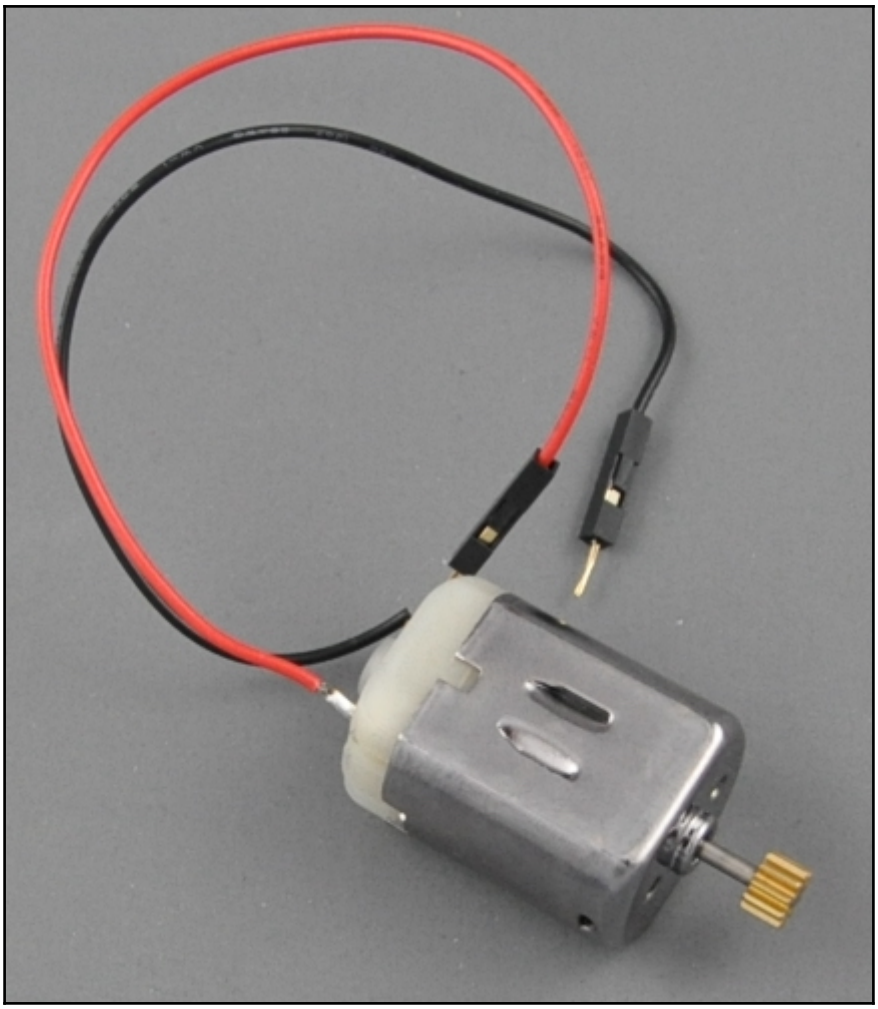

Azure IoT
Cloud
Robot Arm
located in remote
factory
Progress and status
of the Robot Arm is monitored
and managed through a cell
phone app

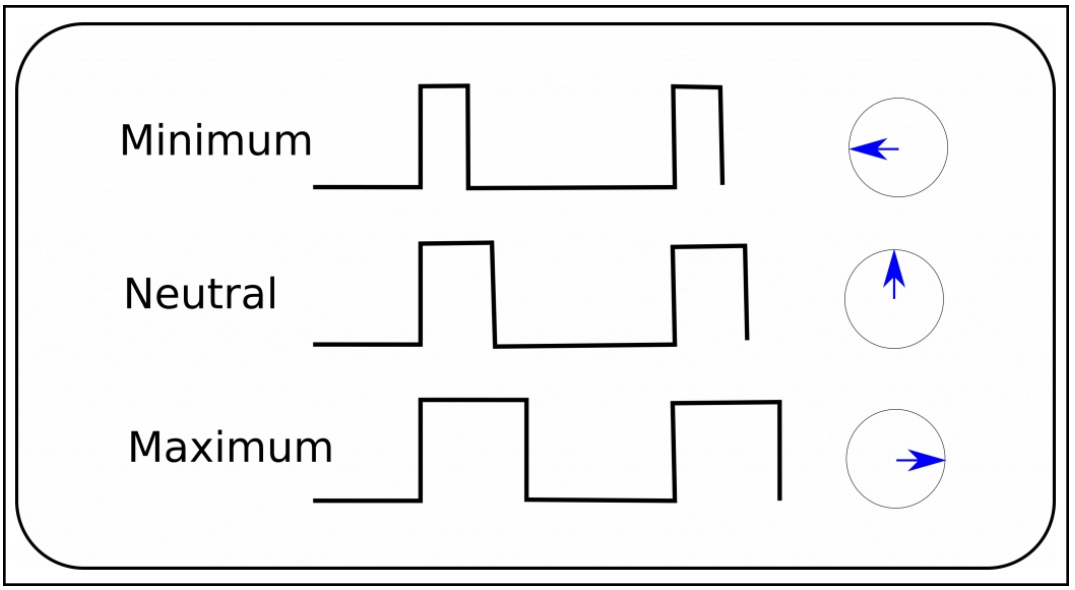

Thonny - /home/pi/...
[pi]
Sense HAT Emulator
Thonny - /home/pi/DisplayWeather.py @ 17 : 32
File Edit View Run Tools Help
CurrentWeather.py
DisplayWeather.py
from sense_emu import SenseHat
from CurrentWeather import CurrentWeather
class DisplayWeather:
current_weather = ''
def __init__(self, current_weather):
self.current_weather = current_weather
def display(self):
sense_hat_emulator = SenseHat()
message = ("%s %sC %s wind speed %s km/h"
%(self.current_weather.getCity(),
self.current_weather.getTemperature(),
self.current_weather.getWeatherConditions(),
self.current_weather.getWindSpeed()))
sense_hat_emulator.show_message(message)
if __name__ == "__main__":
current_weather = CurrentWeather('Toronto')
display_weather = DisplayWeather(current_weather)
display_weather.display()
Shell
>>> %Run DisplayWeather.py
Sense HAT Emulator
File Edit Help
Screen
Temperature
Pressure
Humidity
Orientation
Yaw
Pitch
Roll
0.0°
25.0°C
1013.0mbar
45.0%
Joystick

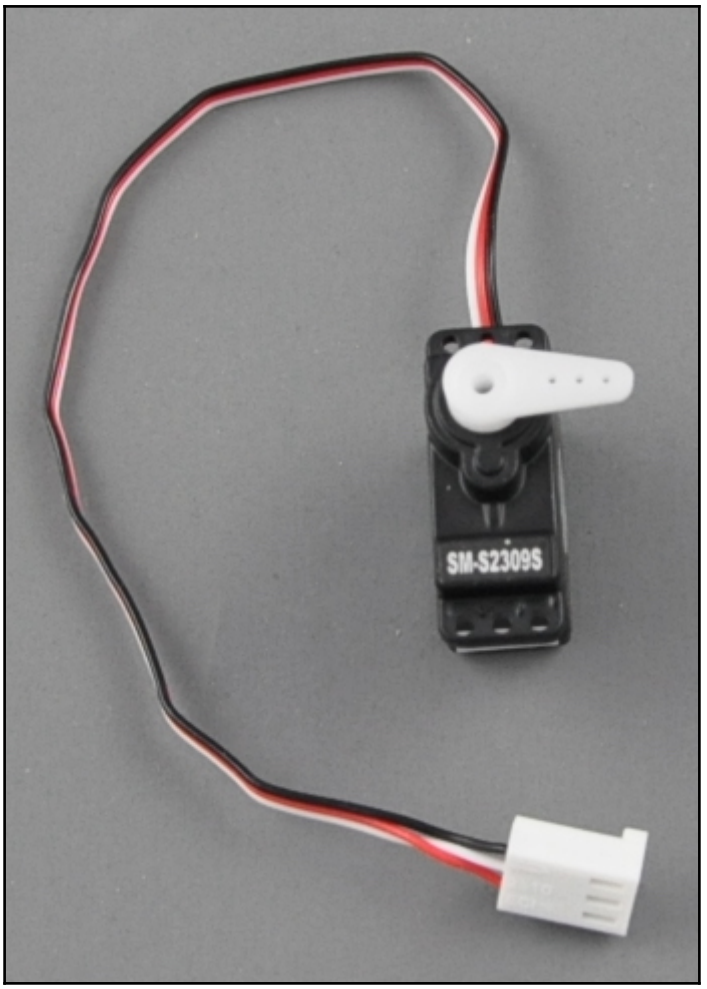
Chapter 14: Controlling a Servo with Python



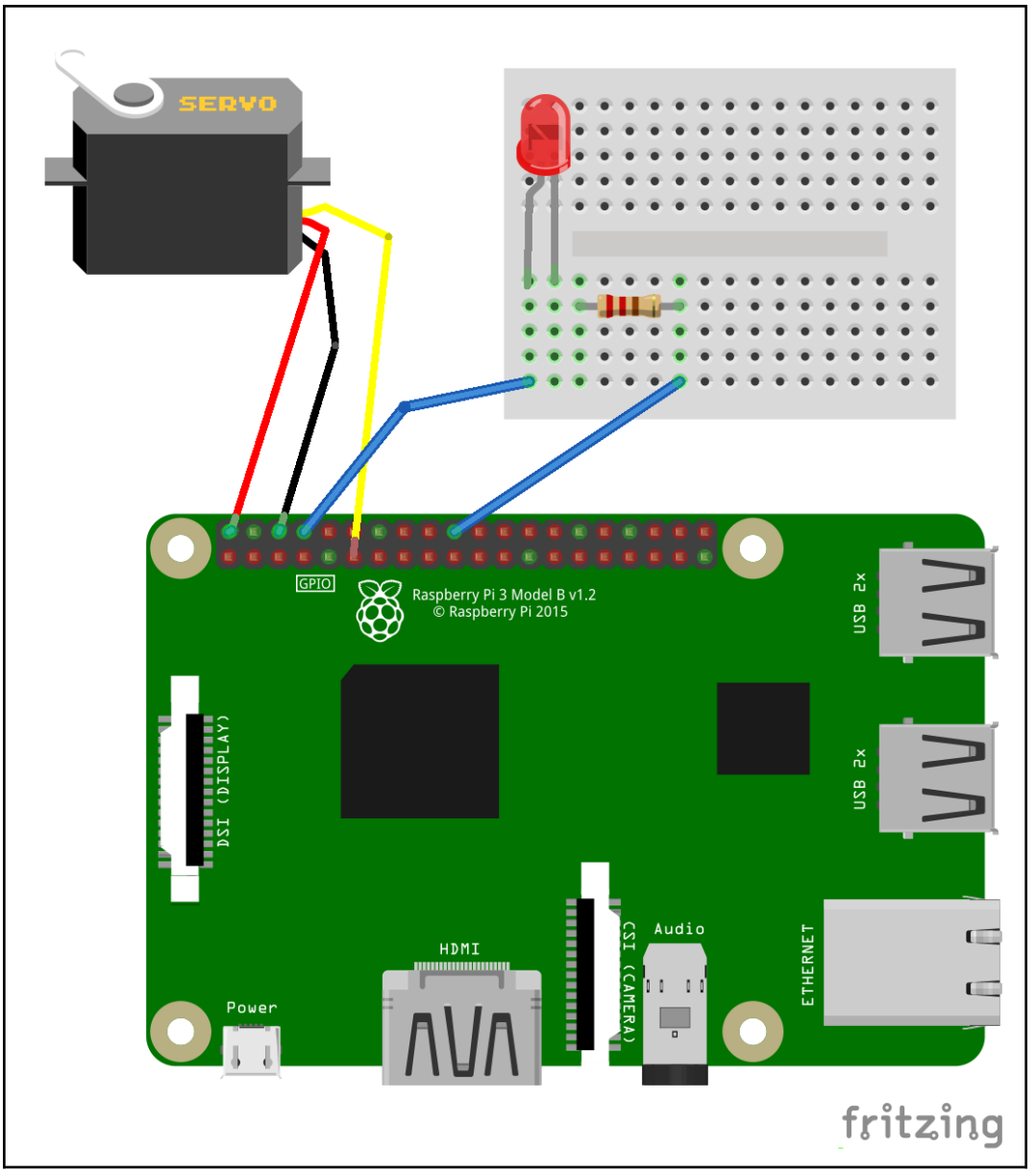


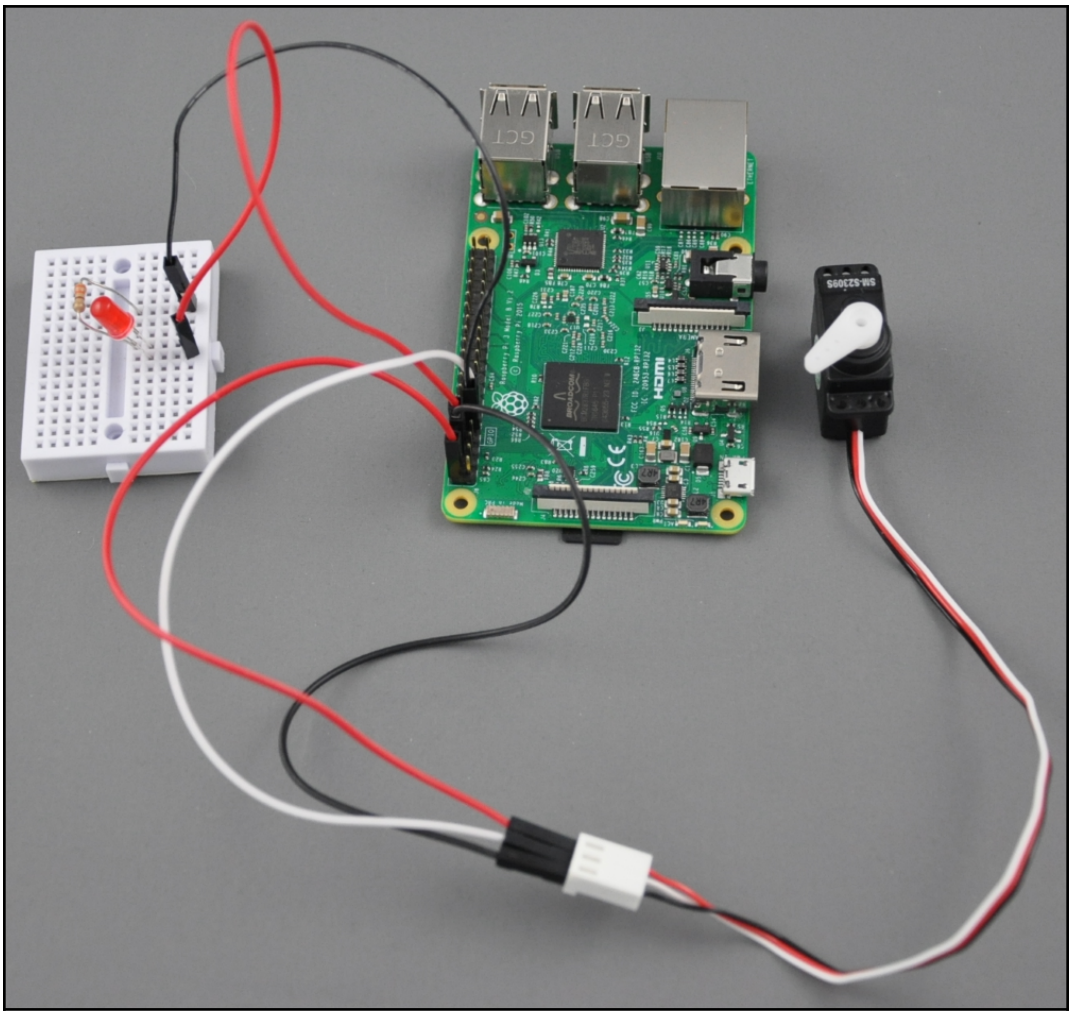


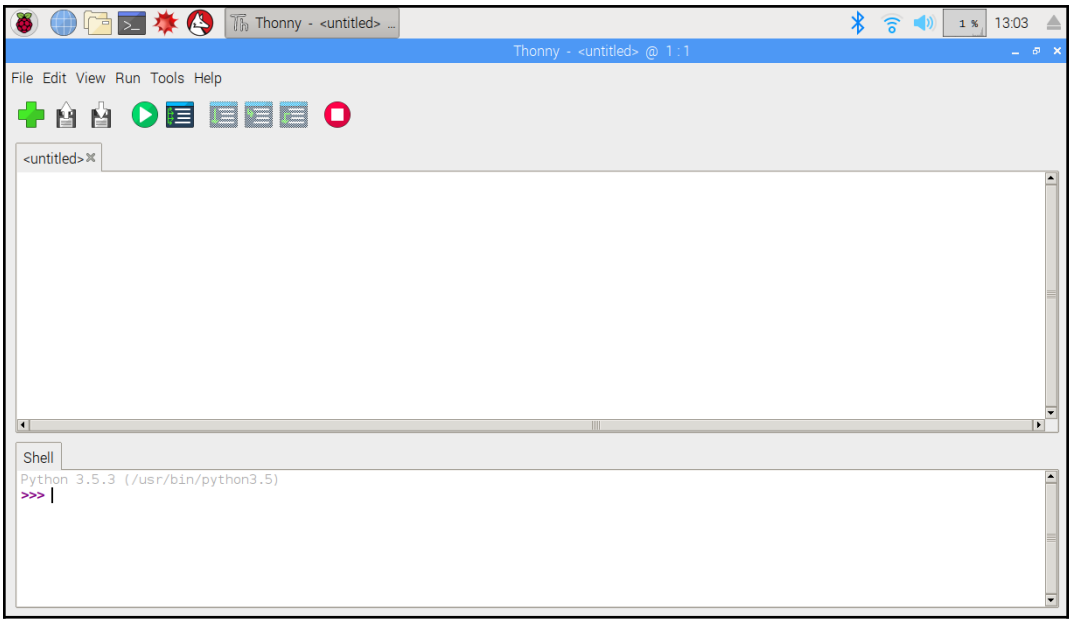

Minimum
Neutral
Maximum


SM-S2309S

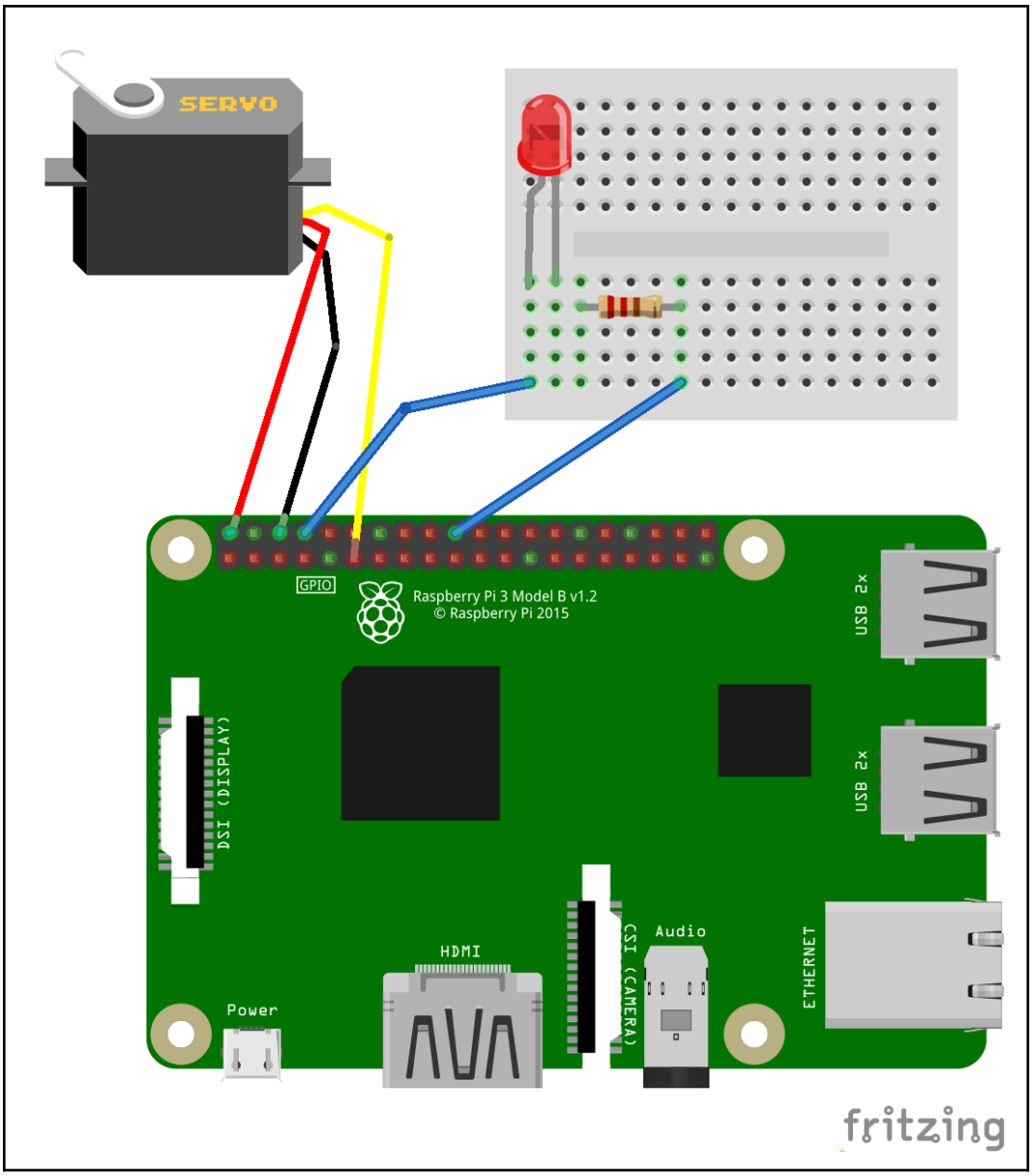
Power / Positive (+)	Signal / Data	Ground (-)
Red	Yellow	Black
Red	White	Black
Red	Orange	Black
Red	Orange	Brown
Red	Blue	Black

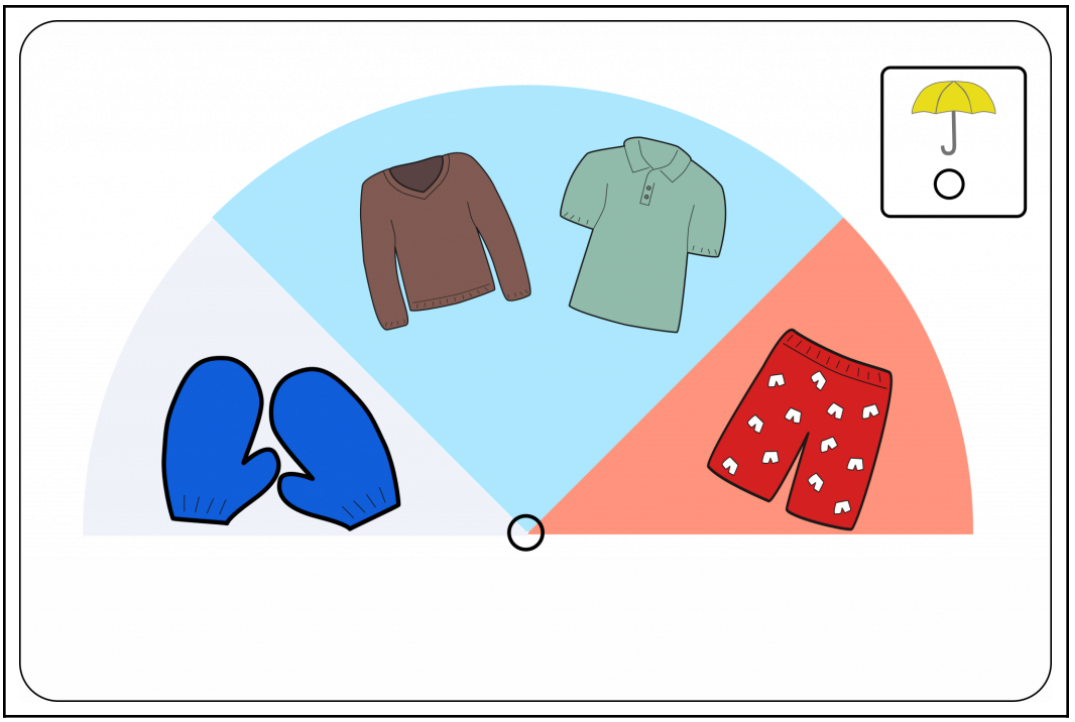

SERVO
GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing

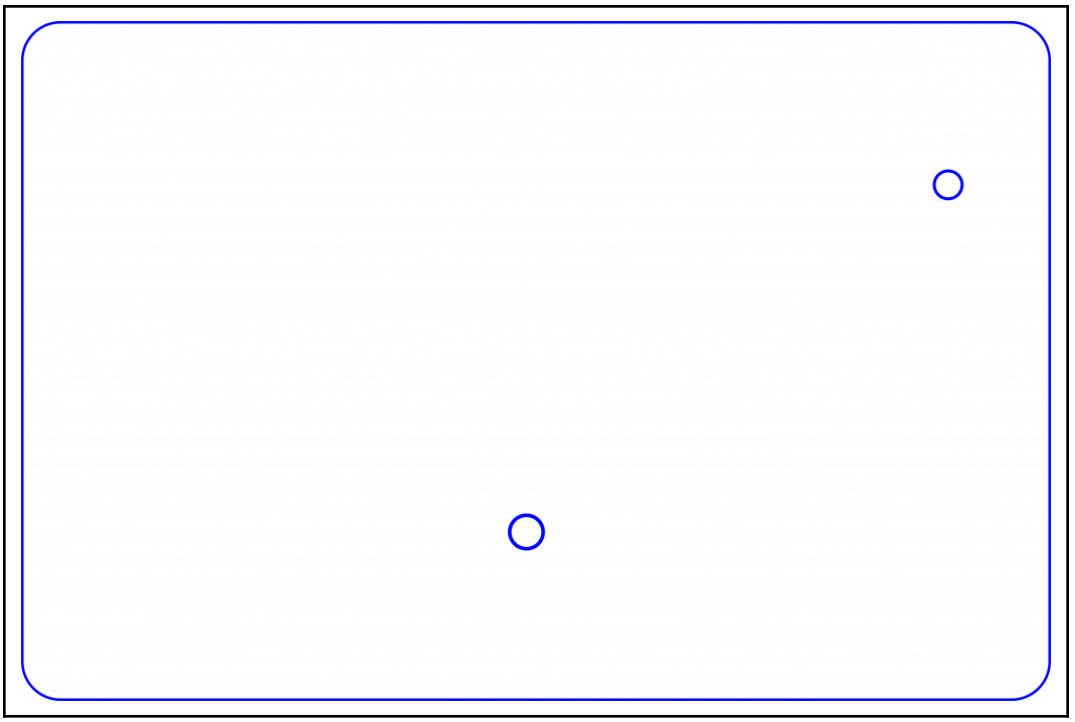


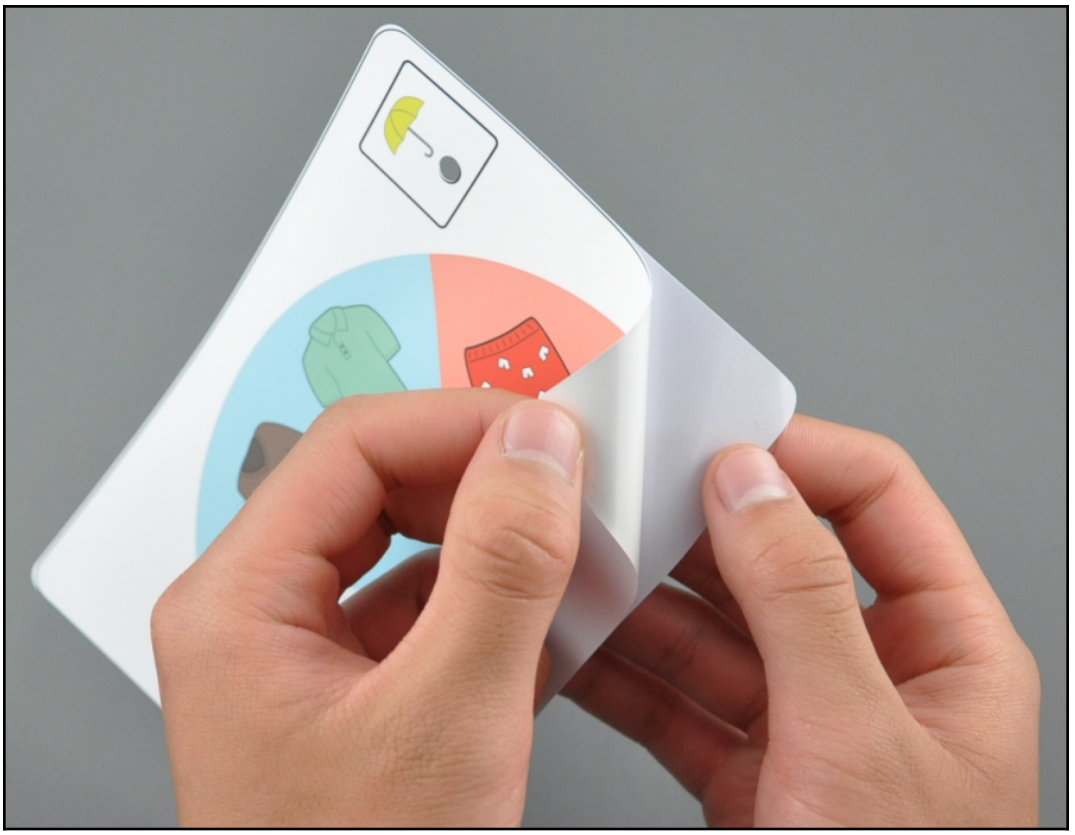

Thonny - <untitled> ...
1 %
13:03
Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
<untitled>
Shell
Python 3.5.3 (/usr/bin/python3.5)
>>>

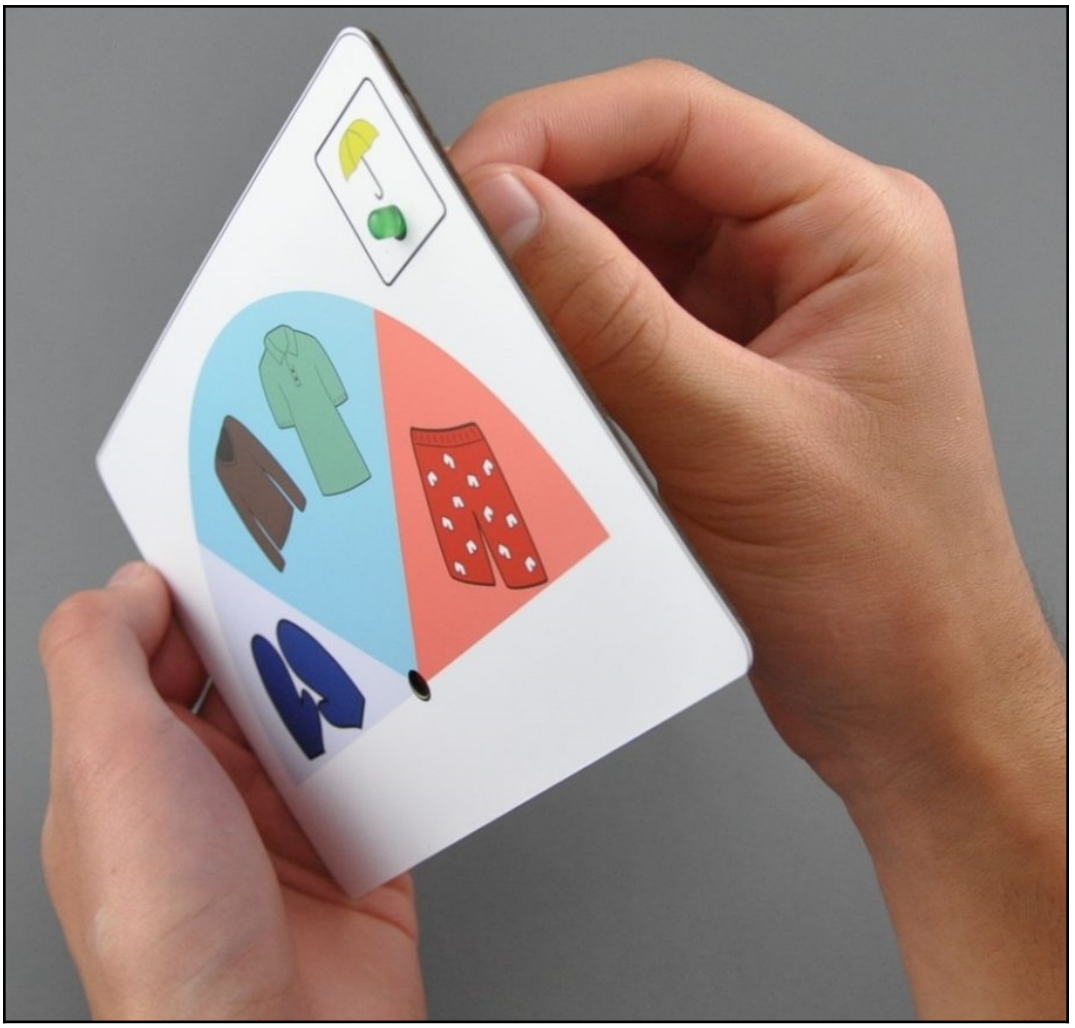
Chapter 15: Working with the Servo Control Code to Control an Analog Device

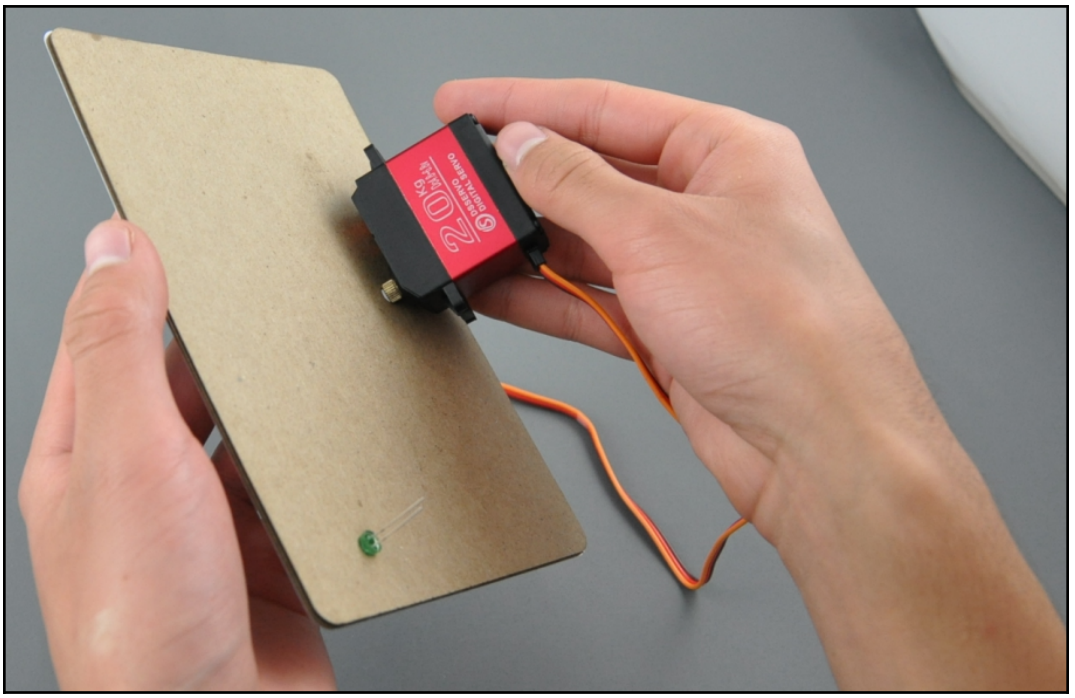

SERVO
GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing

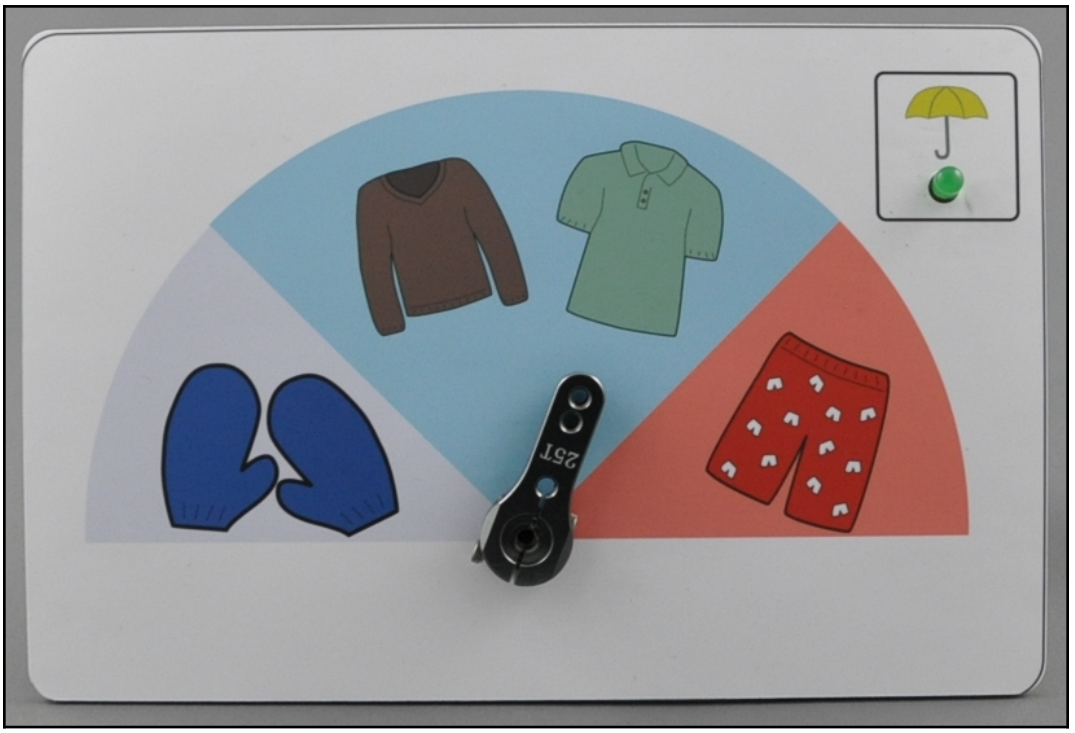




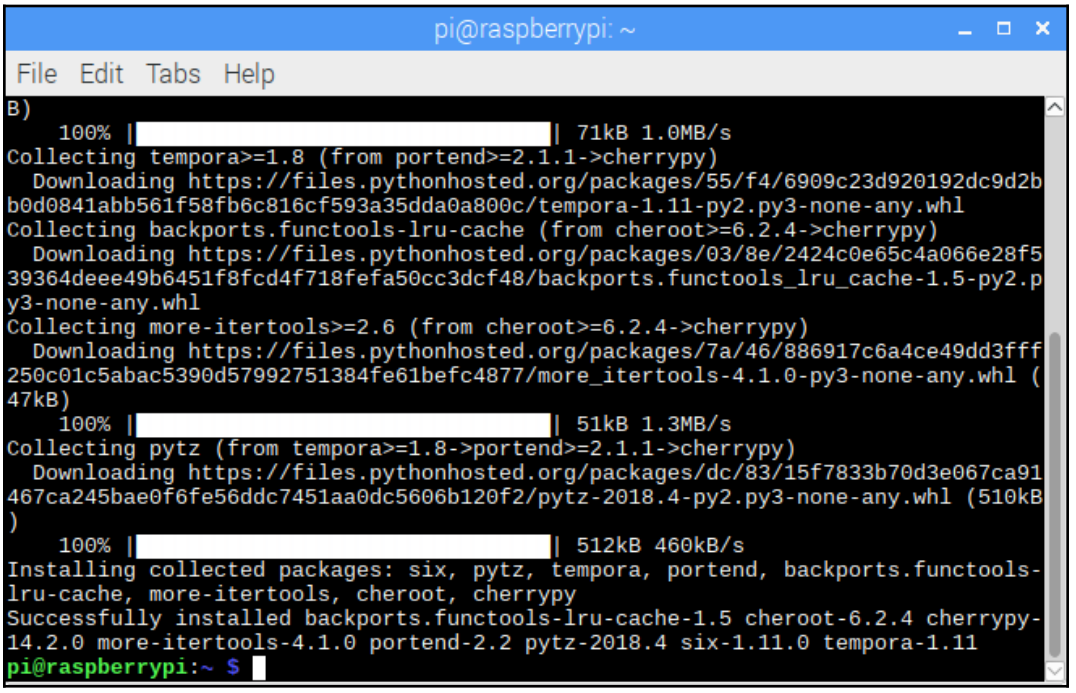


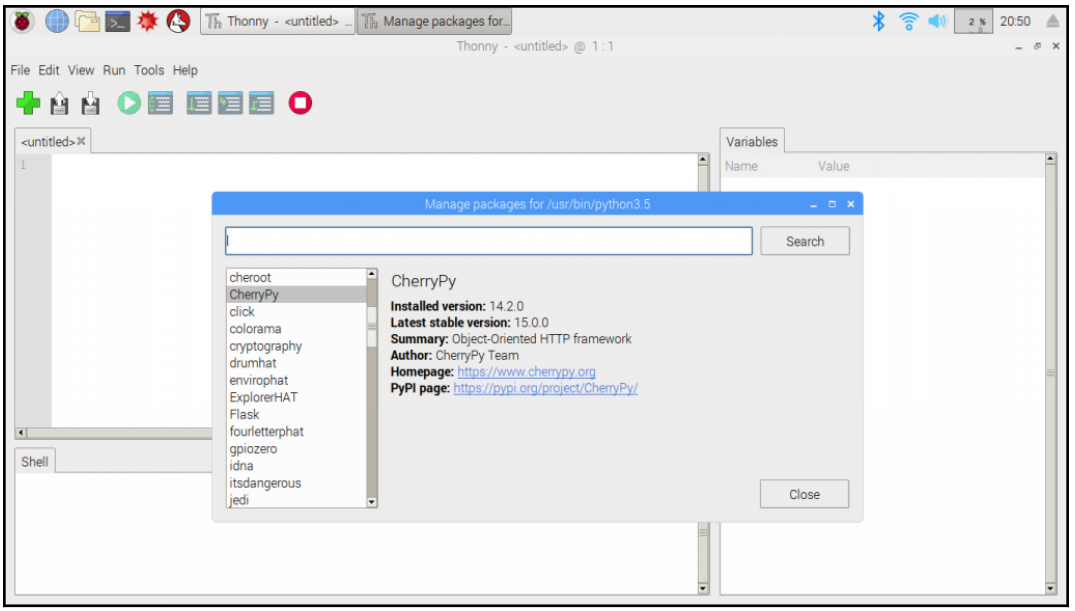


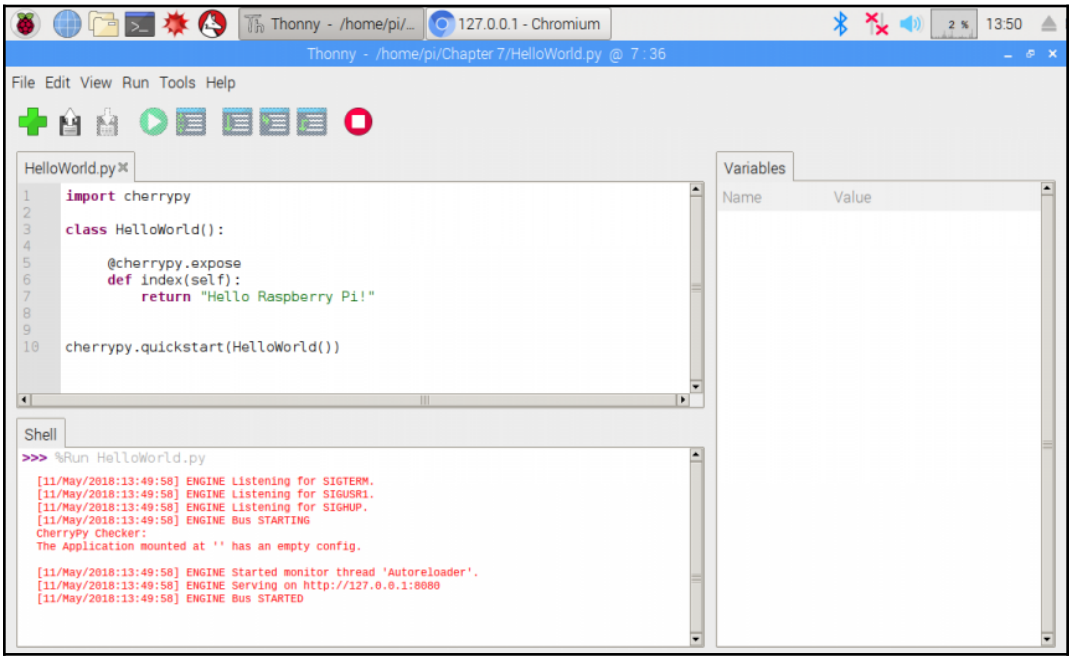

20Kg
DIGITAL SERVO

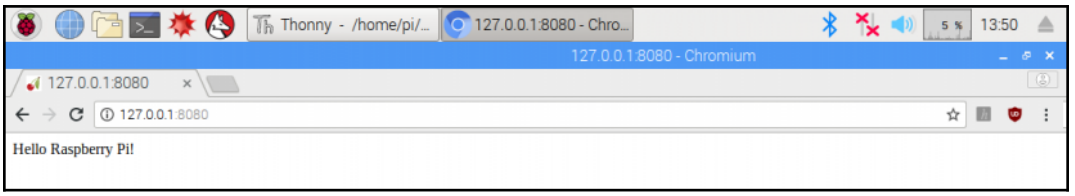

25T

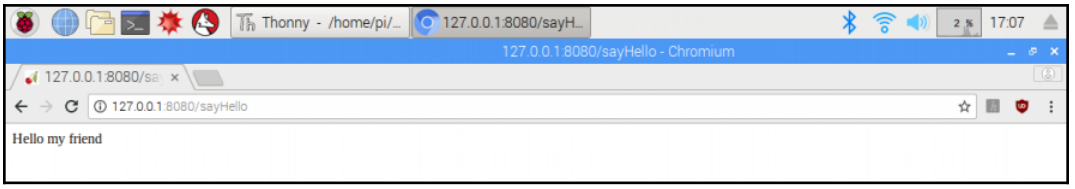
Chapter 16: Setting Up a Raspberry Pi Web Server



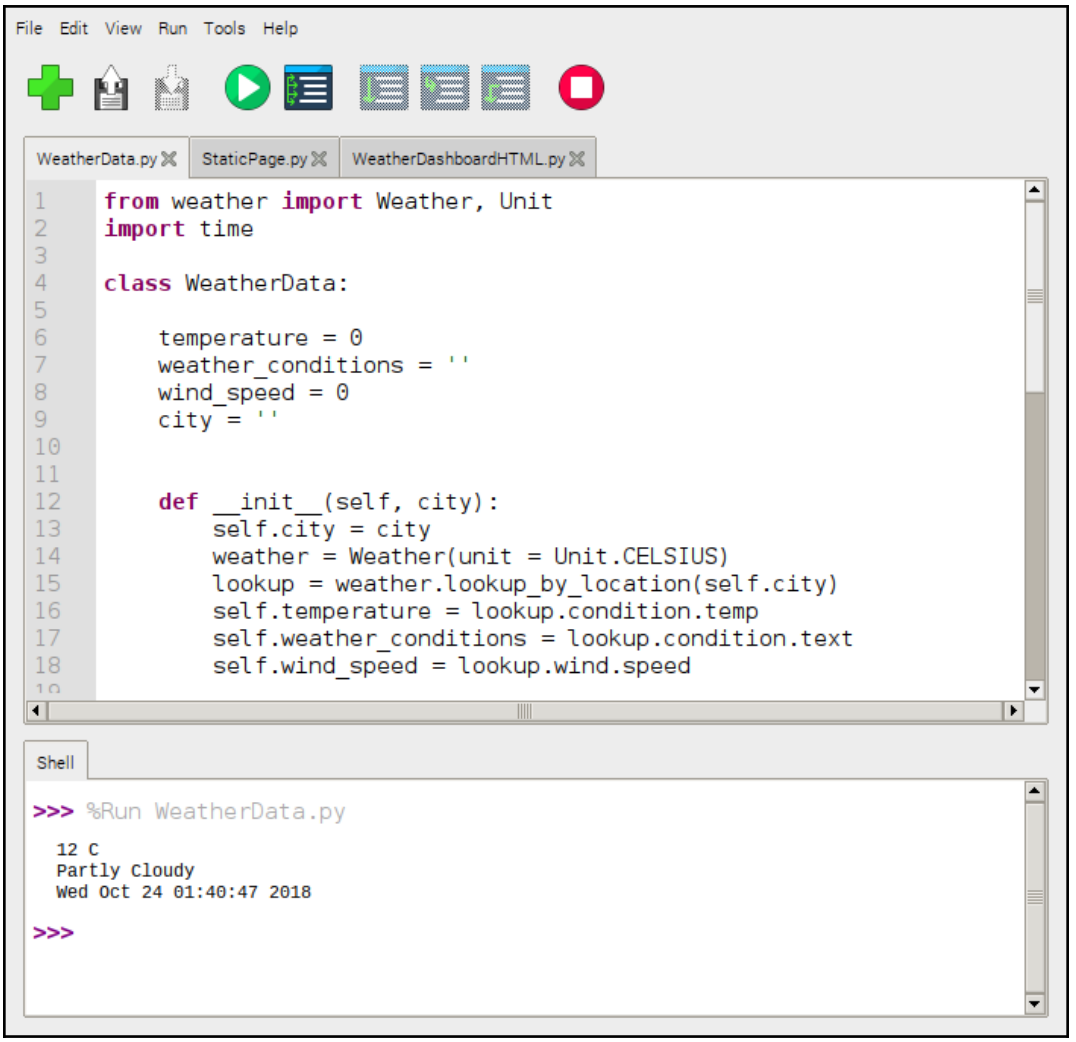

Thonny - <untitled> ...
Manage packages for...
20:50
Thonny - <untitled> @ 1 : 1
File Edit View Run Tools Help
<untitled>
Variables
Name Value
Manage packages for /usr/bin/python3.5
Search
cheroot
CherryPy
click
colorama
cryptography
drumhat
envirophat
ExplorerHAT
Flask
fourletterphat
gpiozero
idna
itsdangerous
jedi
CherryPy
Installed version: 14.2.0
Latest stable version: 15.0.0
Summary: Object-Oriented HTTP framework
Author: CherryPy Team
Homepage: https://www.cherrypy.org
PyPI page: https://pypi.org/project/CherryPy/
Close
Shell

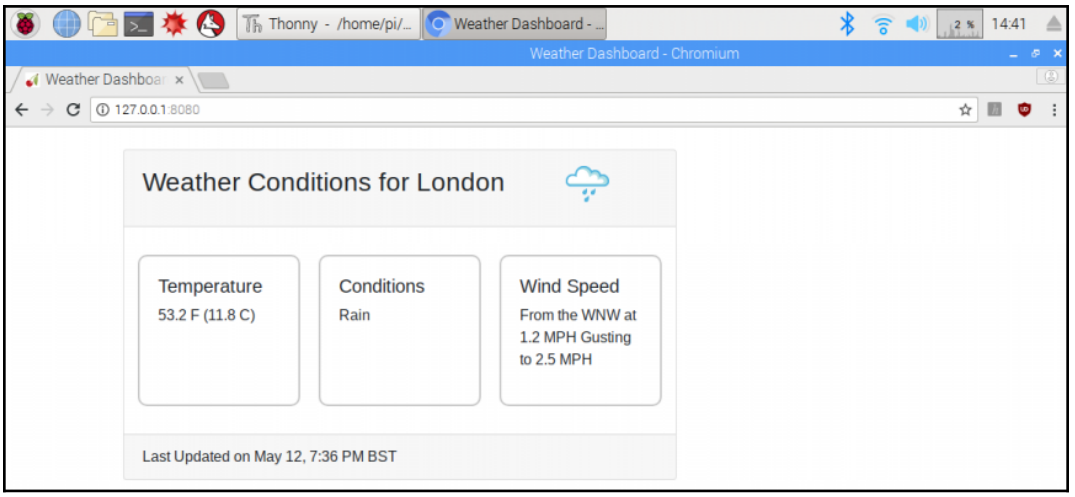

Thonny - /home/pi/...
127.0.0.1 - Chromium
13:50
Thonny - /home/pi/Chapter 7/HelloWorld.py @ 7 : 36
File Edit View Run Tools Help
HelloWorld.py
import cherrypy
class HelloWorld():
@cherrypy.expose
def index(self):
return "Hello Raspberry Pi!"
cherrypy.quickstart(HelloWorld())
Variables
Name Value
Shell
>>> %Run HelloWorld.py
[11/May/2018:13:49:58] ENGINE Listening for SIGTERM.
[11/May/2018:13:49:58] ENGINE Listening for SIGUSR1.
[11/May/2018:13:49:58] ENGINE Listening for SIGHUP.
[11/May/2018:13:49:58] ENGINE Bus STARTING
CherryPy Checker:
The Application mounted at '' has an empty config.
[11/May/2018:13:49:58] ENGINE Started monitor thread 'Autoreloader'.
[11/May/2018:13:49:58] ENGINE Serving on http://127.0.0.1:8080
[11/May/2018:13:49:58] ENGINE Bus STARTED


Thonny - /home/pi/...
127.0.0.1:8080 - Chro...
13:50
127.0.0.1:8080 - Chromium
127.0.0.1:8080
127.0.0.1:8080
Hello Raspberry Pi!

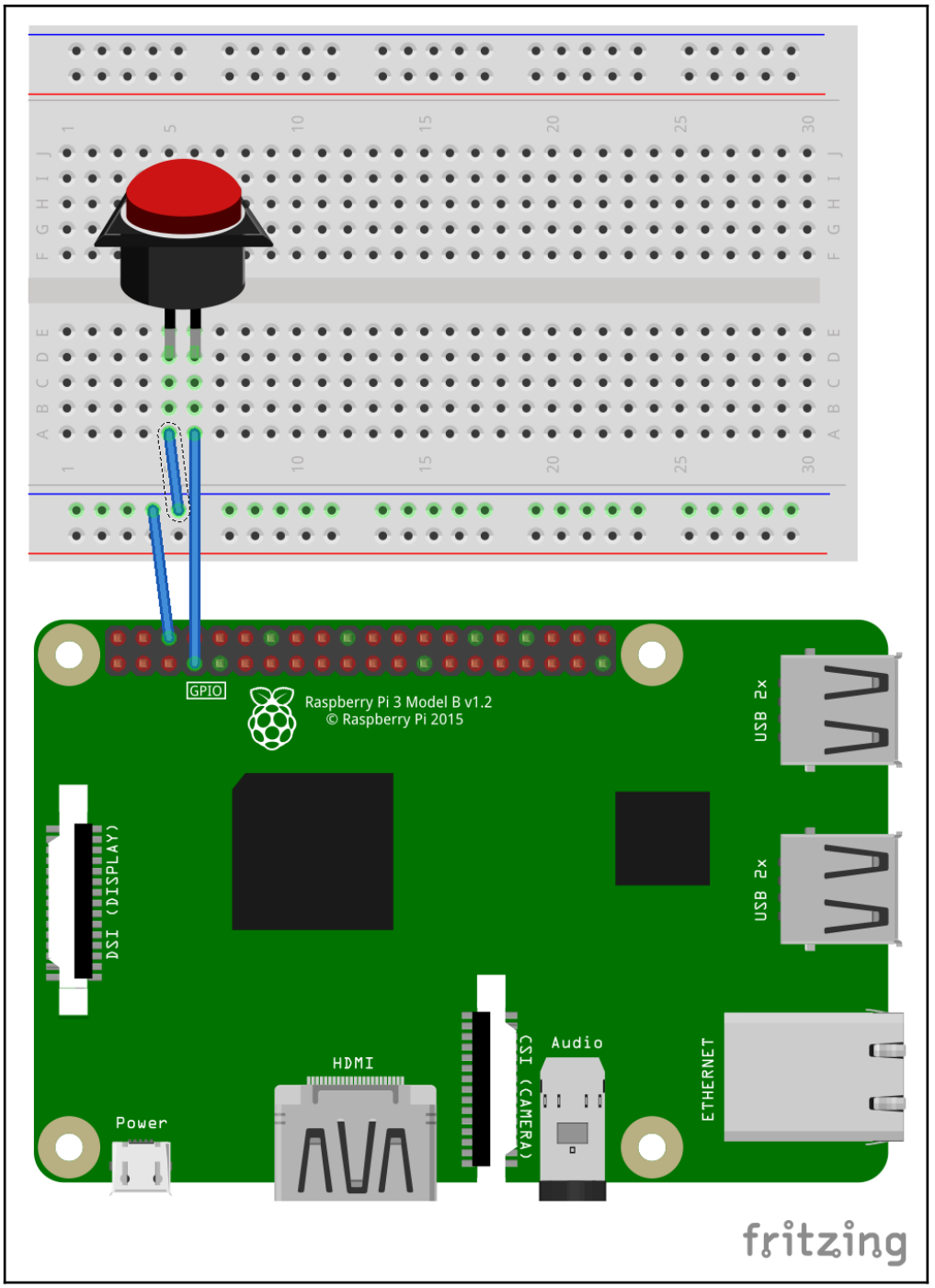

Thonny - /home/pi/...
127.0.0.1:8080/sayH...
17:07
127.0.0.1:8080/sayHello - Chromium
127.0.0.1:8080/sa
127.0.0.1:8080/sayHello
Hello my friend

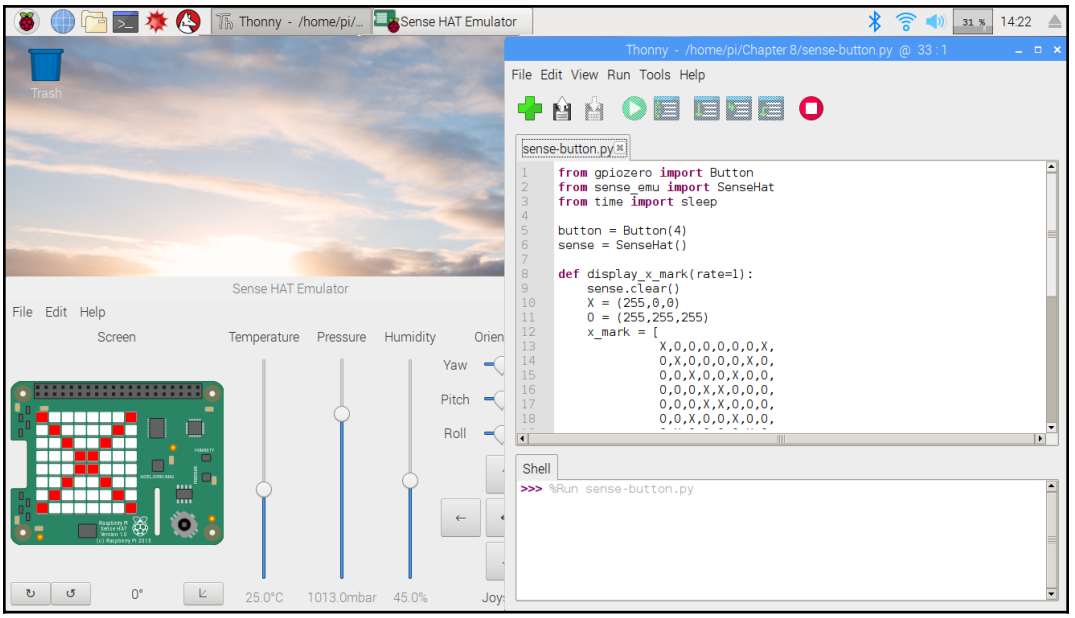

127.0.0.1:8080
127.0.0.1:8080
This is a static HTML page.

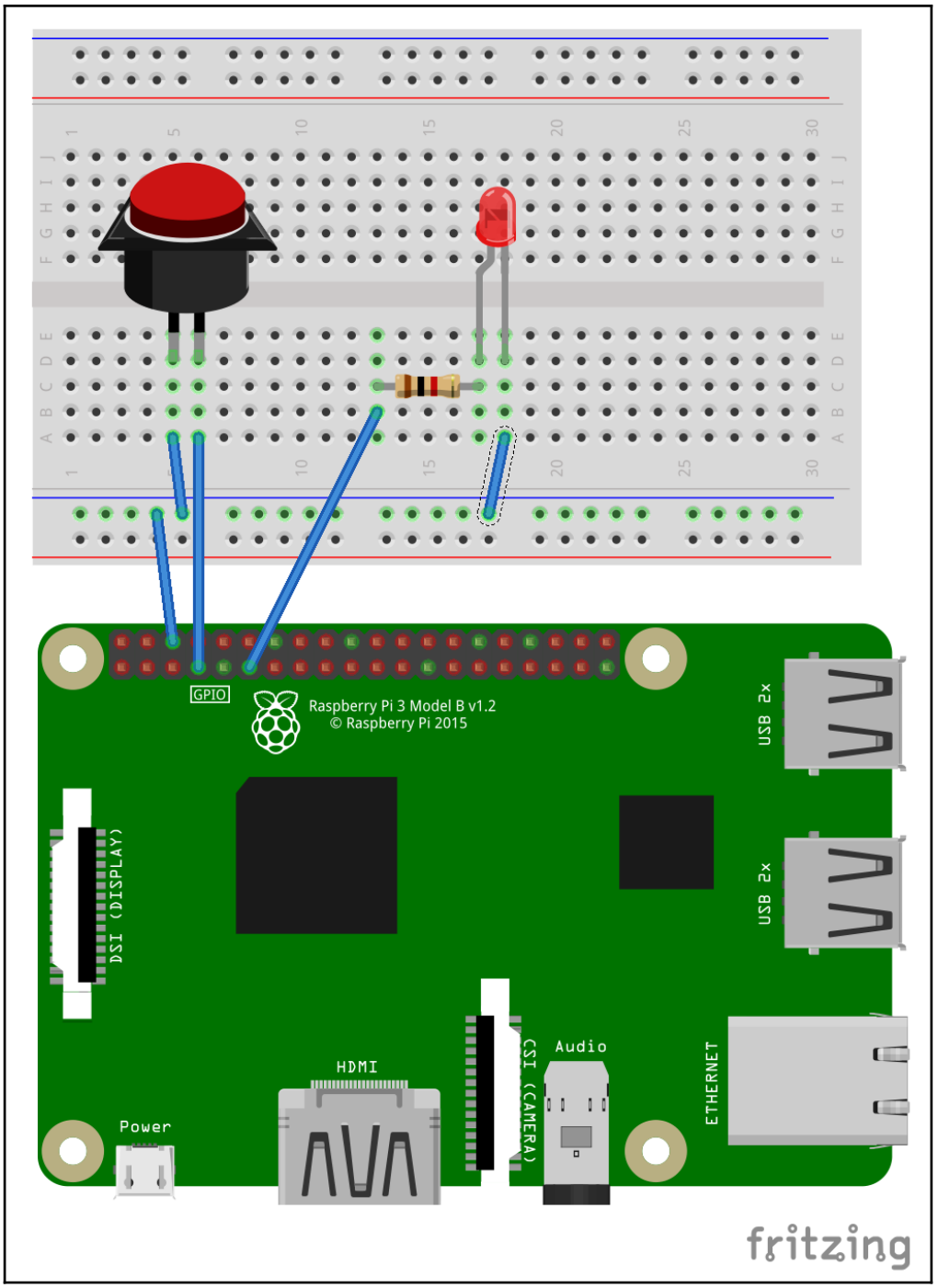

File Edit View Run Tools Help
WeatherData.py
StaticPage.py
WeatherDashboardHTML.py
1 from weather import Weather, Unit
2 import time
3
4 class WeatherData:
5
6 temperature = 0
7 weather_conditions = ''
8 wind_speed = 0
9 city = ''
10
11
12 def __init__(self, city):
13 self.city = city
14 weather = Weather(unit = Unit.CELSIUS)
15 lookup = weather.lookup_by_location(self.city)
16 self.temperature = lookup.condition.temp
17 self.weather_conditions = lookup.condition.text
18 self.wind_speed = lookup.wind.speed
Shell
>>> %Run WeatherData.py
12 C
Partly Cloudy
Wed Oct 24 01:40:47 2018
>>>

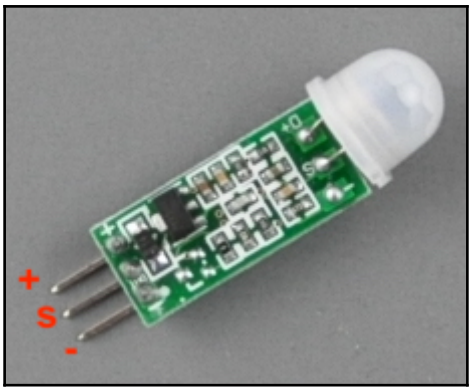

Thonny - /home/pi/...
Weather Dashboard - ...
14:41
Weather Dashboard - Chromium
Weather Dashboa
127.0.0.1:8080
Weather Conditions for London
Temperature
53.2 F (11.8 C)
Conditions
Rain
Wind Speed
From the WNW at 1.2 MPH Gusting to 2.5 MPH
Last Updated on May 12, 7:36 PM BST

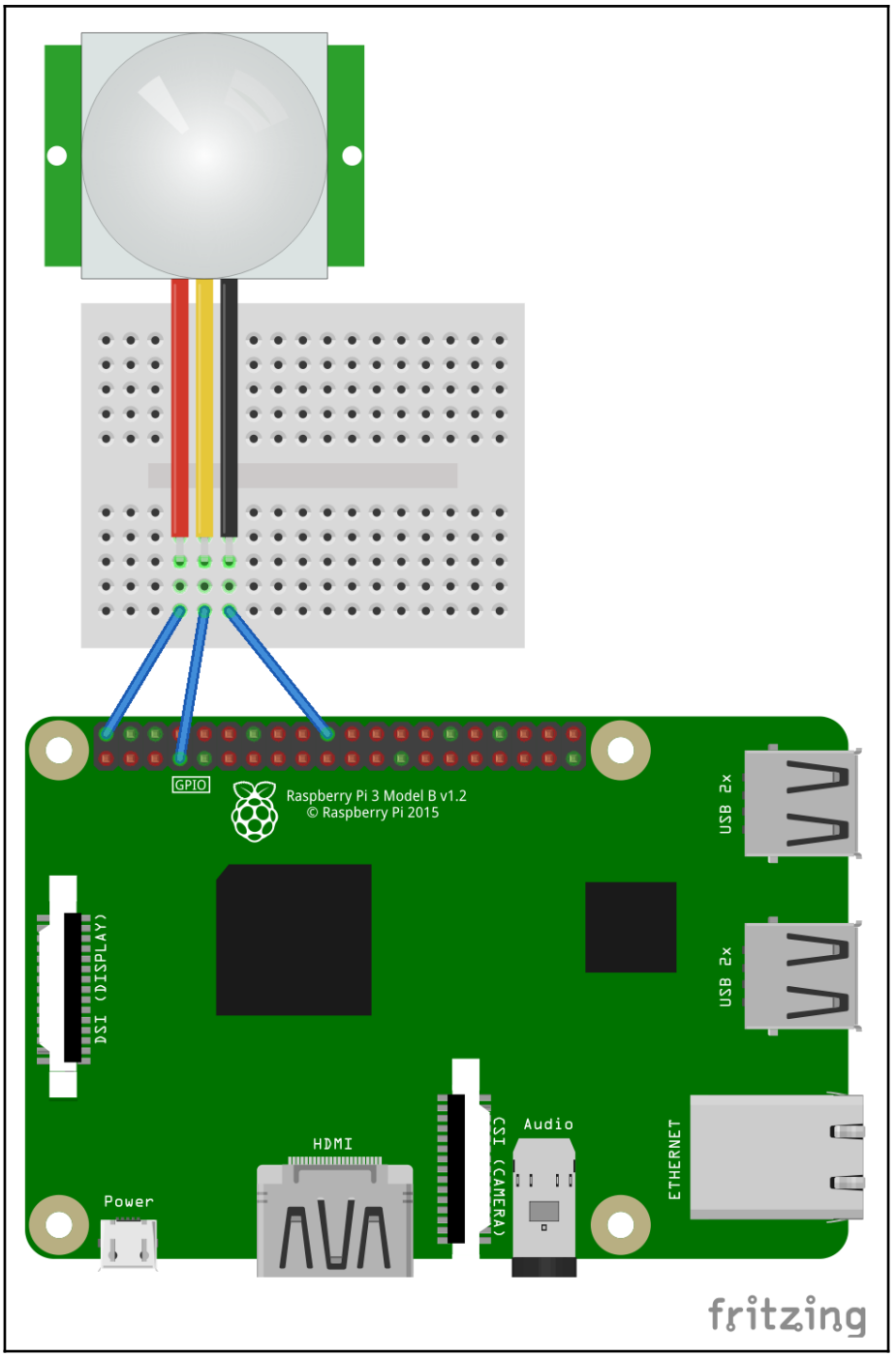
Chapter 17: Reading Raspberry Pi GPIO Sensor Data Using Python

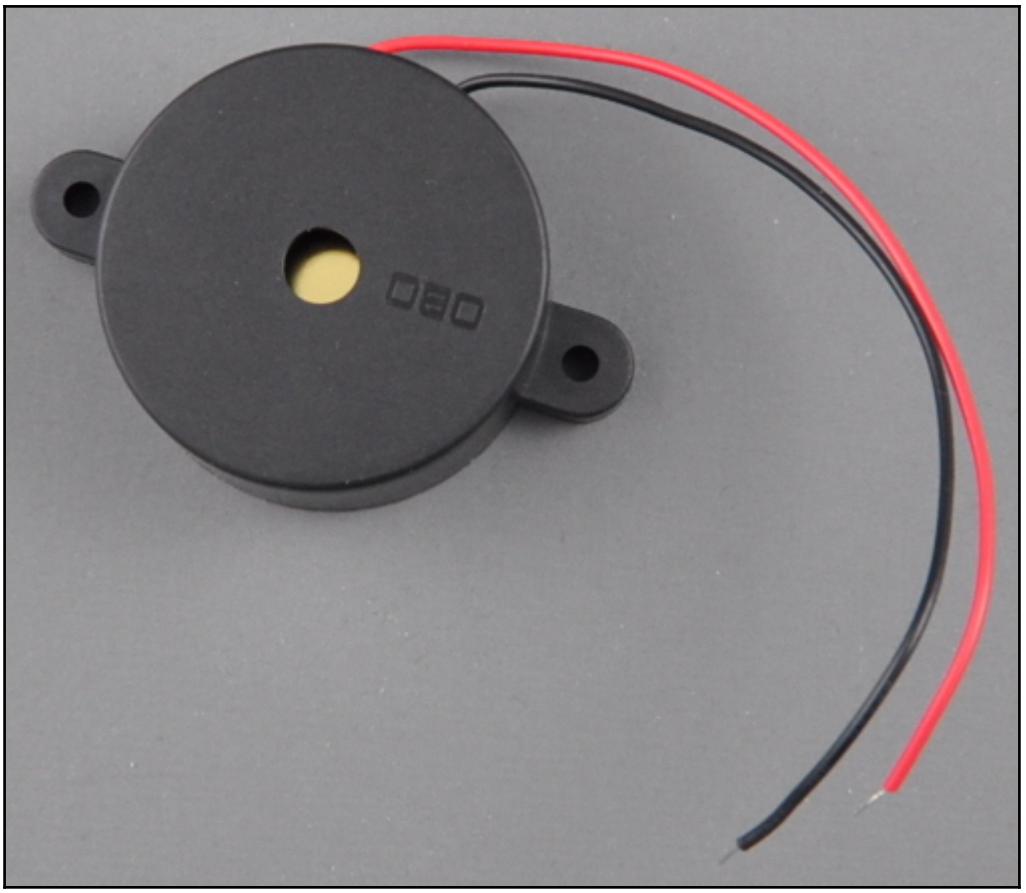

GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing

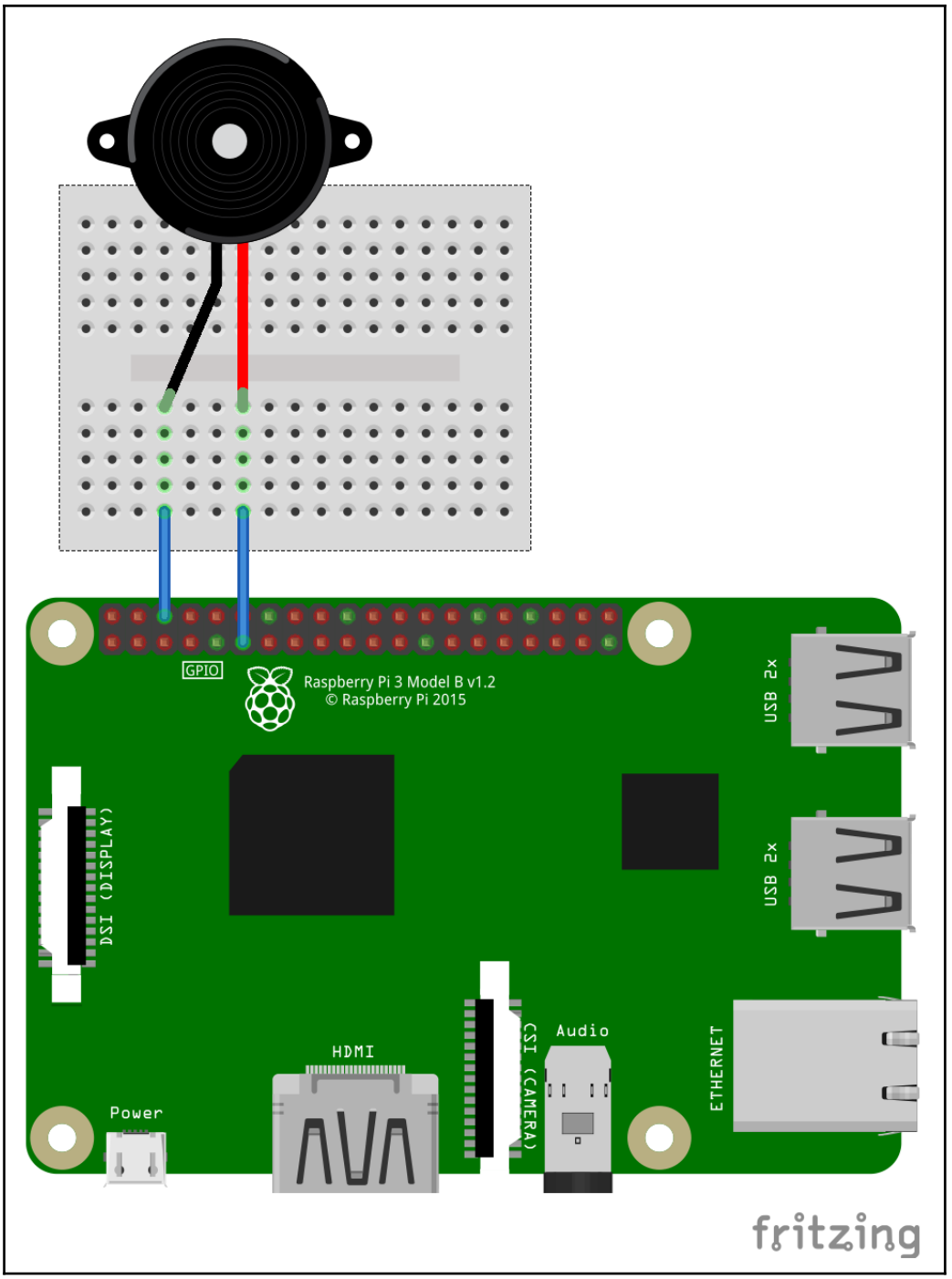

Thonny - /home/pi/...
Sense HAT Emulator
31 %
14:22
Trash
Thonny - /home/pi/Chapter 8/sense-button.py @ 33 : 1
File Edit View Run Tools Help
sense-button.py
from gpiozero import Button
from sense_emu import SenseHat
from time import sleep
button = Button(4)
sense = SenseHat()
def display_x_mark(rate=1):
sense.clear()
X = (255,0,0)
O = (255,255,255)
x_mark = [
X,O,O,O,O,O,O,X,
O,X,O,O,O,O,X,O,
O,O,X,O,O,X,O,O,
O,O,O,X,X,O,O,O,
O,O,O,X,X,O,O,O,
O,O,X,O,O,X,O,O,
Shell
>>> %Run sense-button.py
Sense HAT Emulator
File Edit Help
Screen
Temperature
Pressure
Humidity
Orien
Yaw
Pitch
Roll
0°
25.0°C
1013.0mbar
45.0%

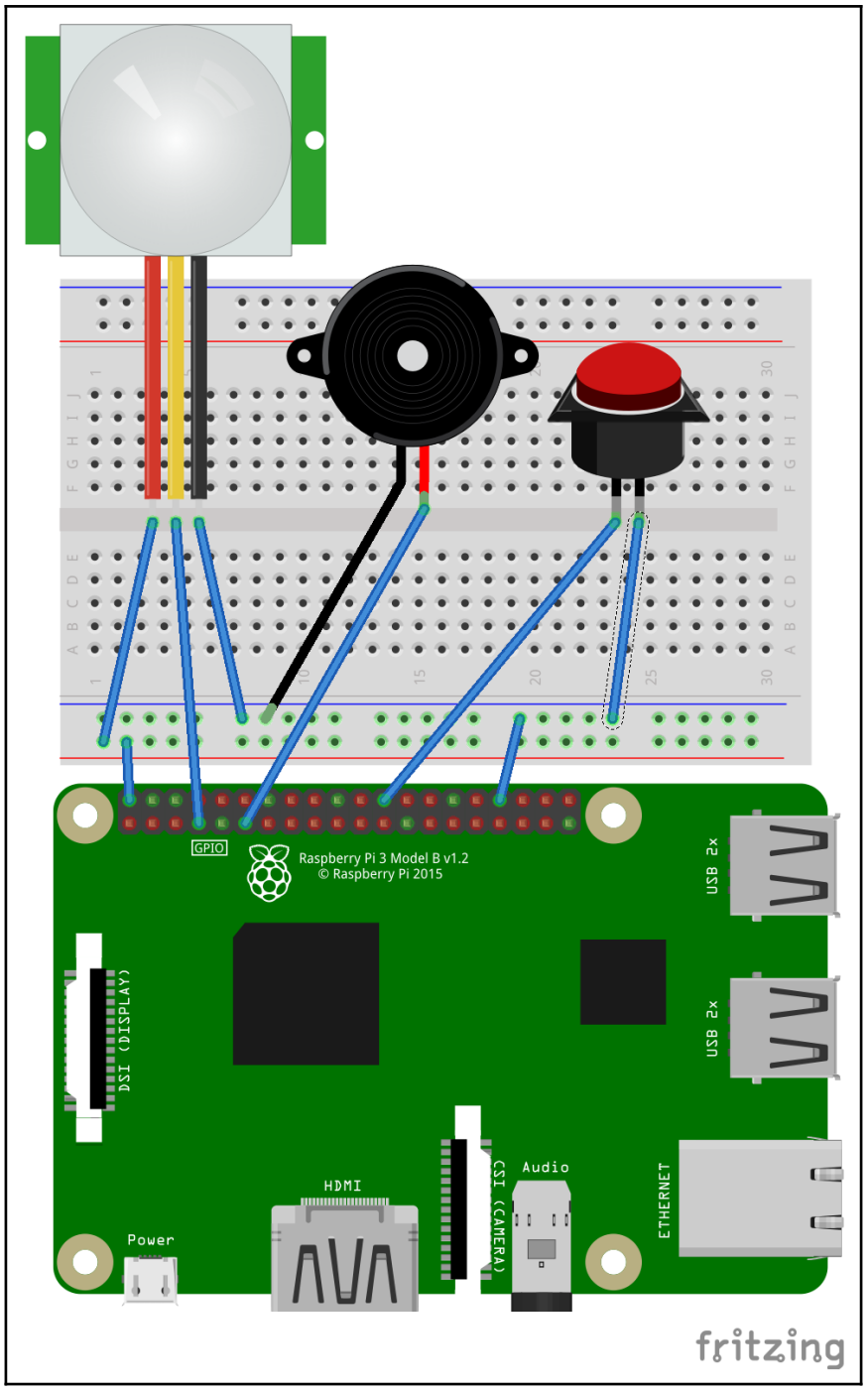

GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing

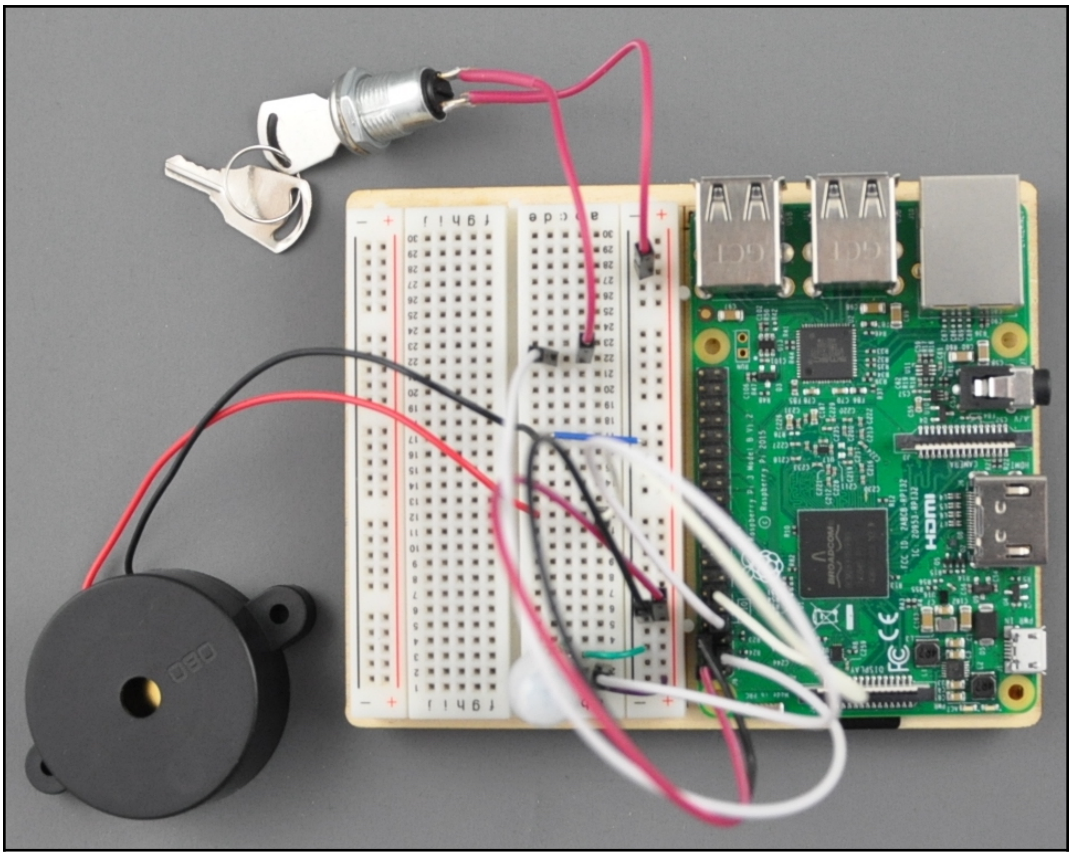

+
S
-

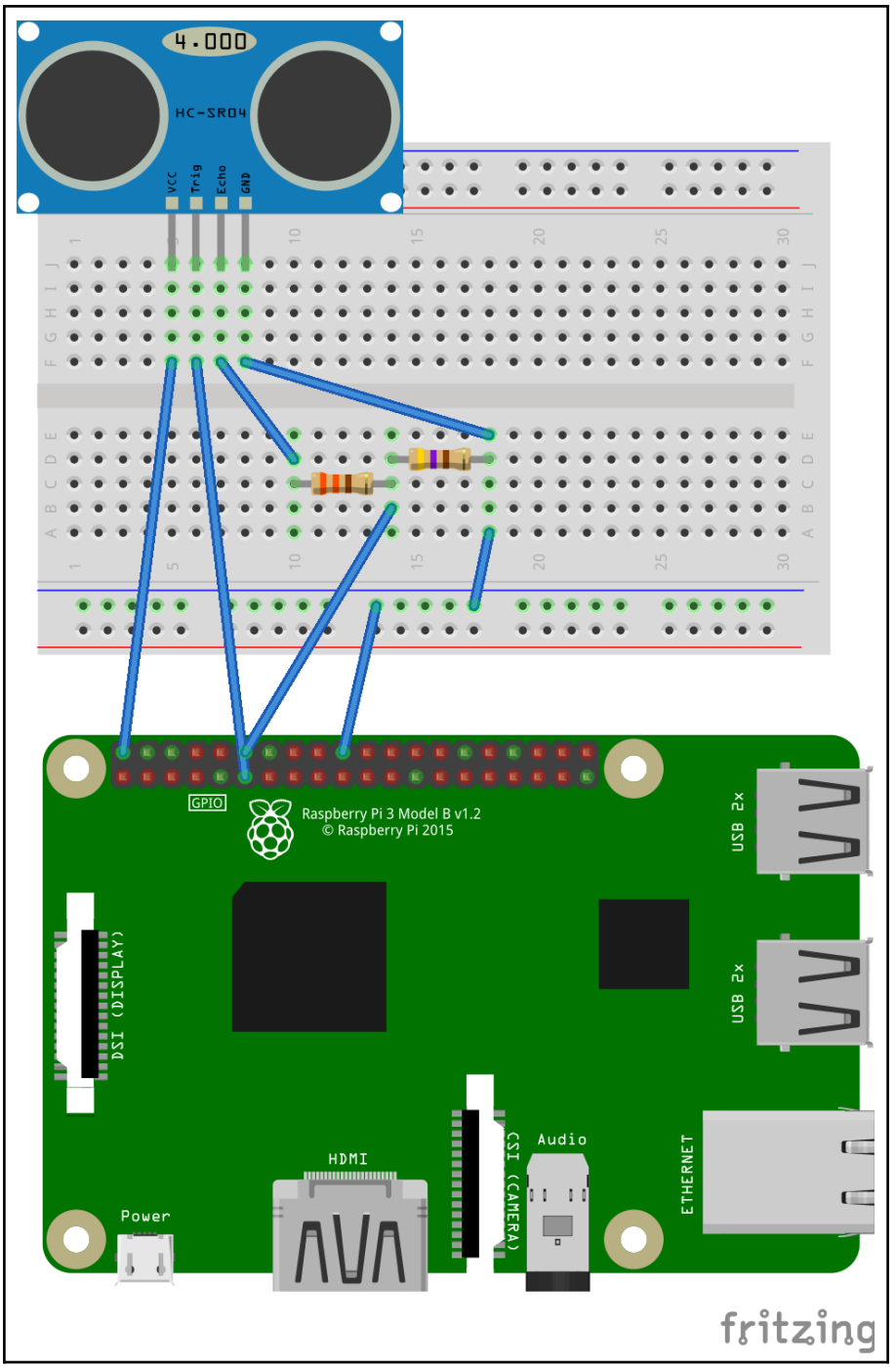

GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing

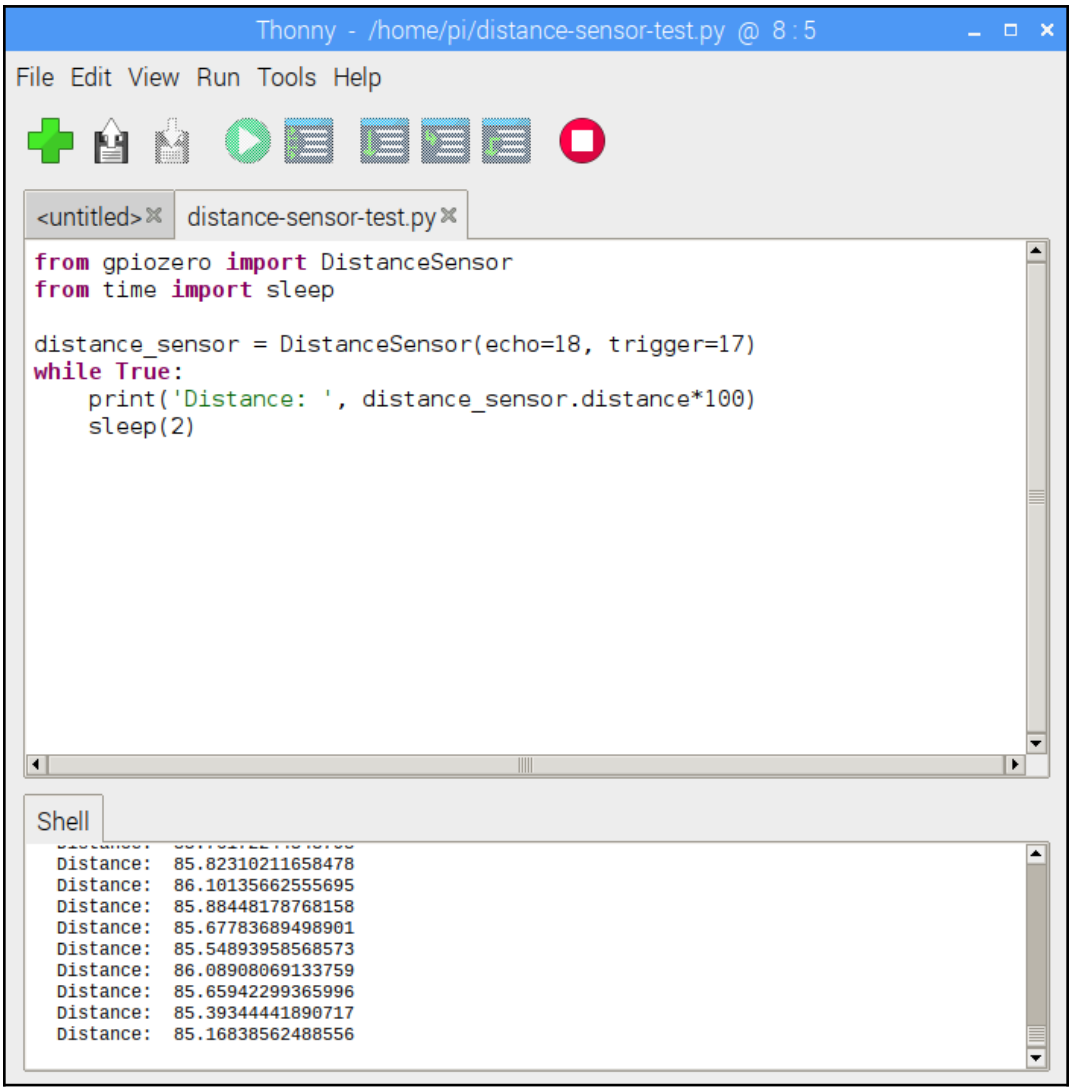


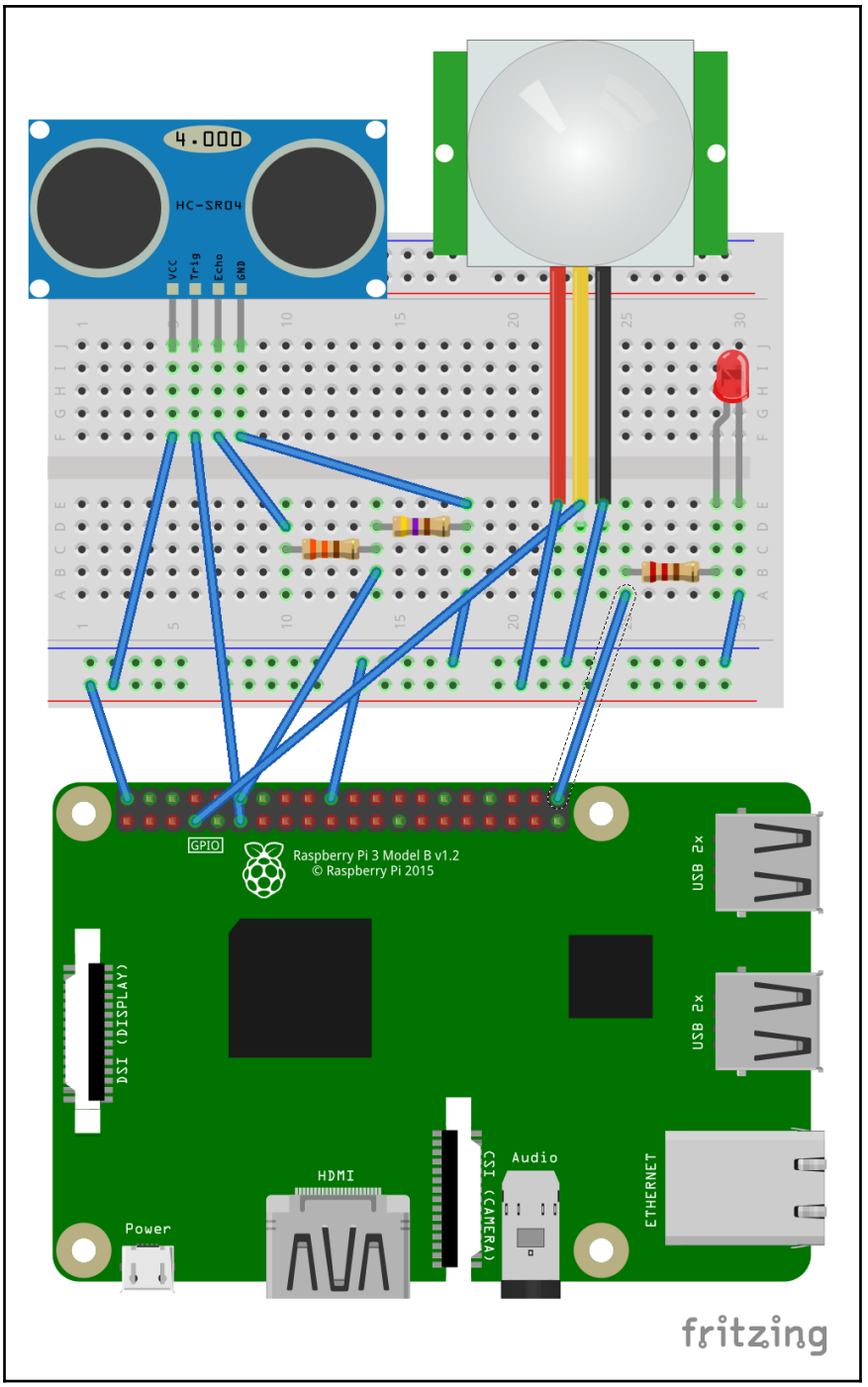

GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing


GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
CSI (CAMERA)
Audio
ETHERNET
HDMI
Power
fritzing

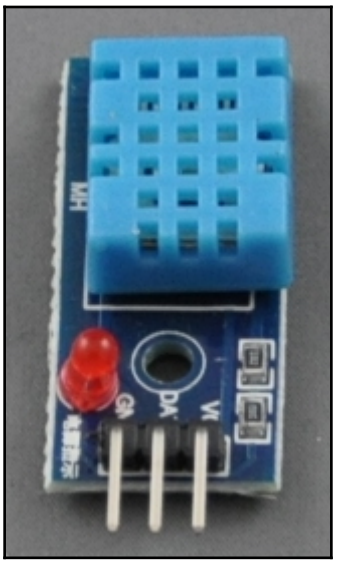


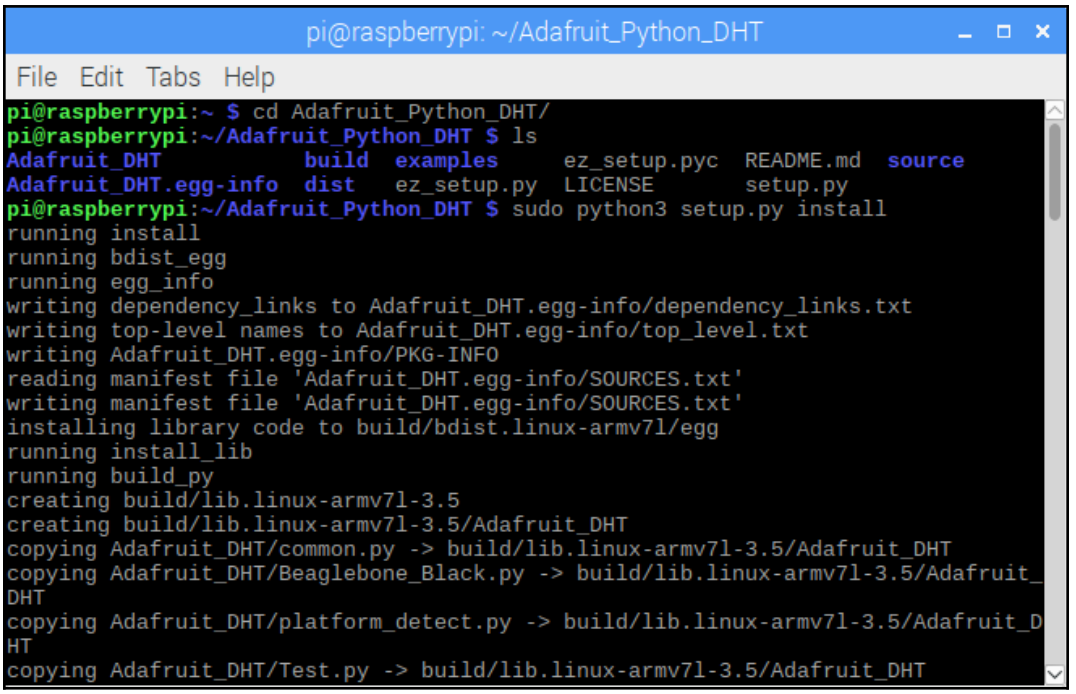

4.000
HC-SR04
VCC
Trig
Echo
GND
GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
CSI (CAMERA)
Audio
HDMI
ETHERNET
Power
fritzing

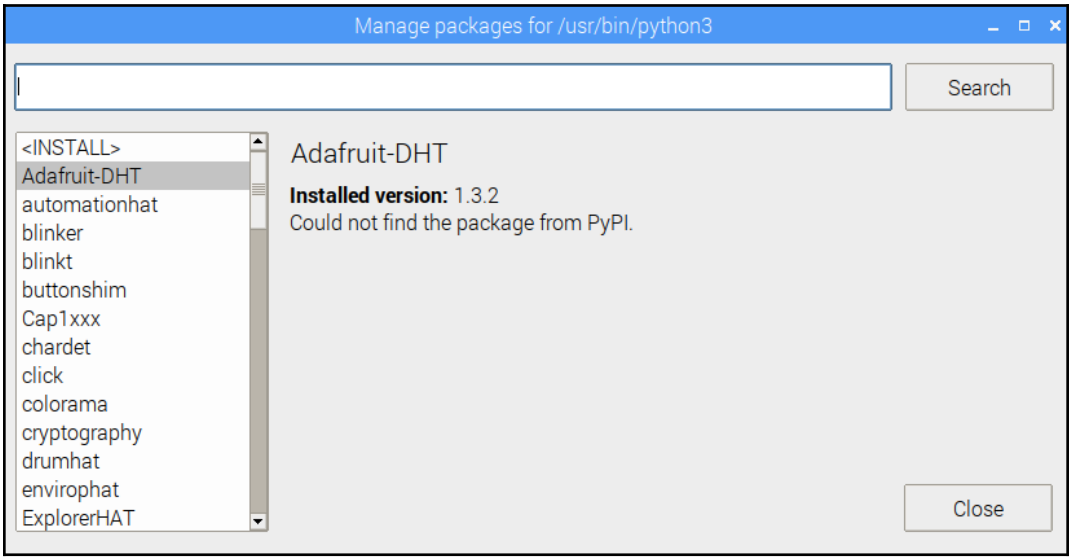

Thonny - /home/pi/distance-sensor-test.py @ 8 : 5
File Edit View Run Tools Help
<untitled>
distance-sensor-test.py
from gpiozero import DistanceSensor
from time import sleep
distance_sensor = DistanceSensor(echo=18, trigger=17)
while True:
print('Distance: ', distance_sensor.distance*100)
sleep(2)
Shell
Distance: 85.82310211658478
Distance: 86.10135662555695
Distance: 85.88448178768158
Distance: 85.67783689498901
Distance: 85.54893958568573
Distance: 86.08908069133759
Distance: 85.65942299365996
Distance: 85.39344441890717
Distance: 85.16838562488556

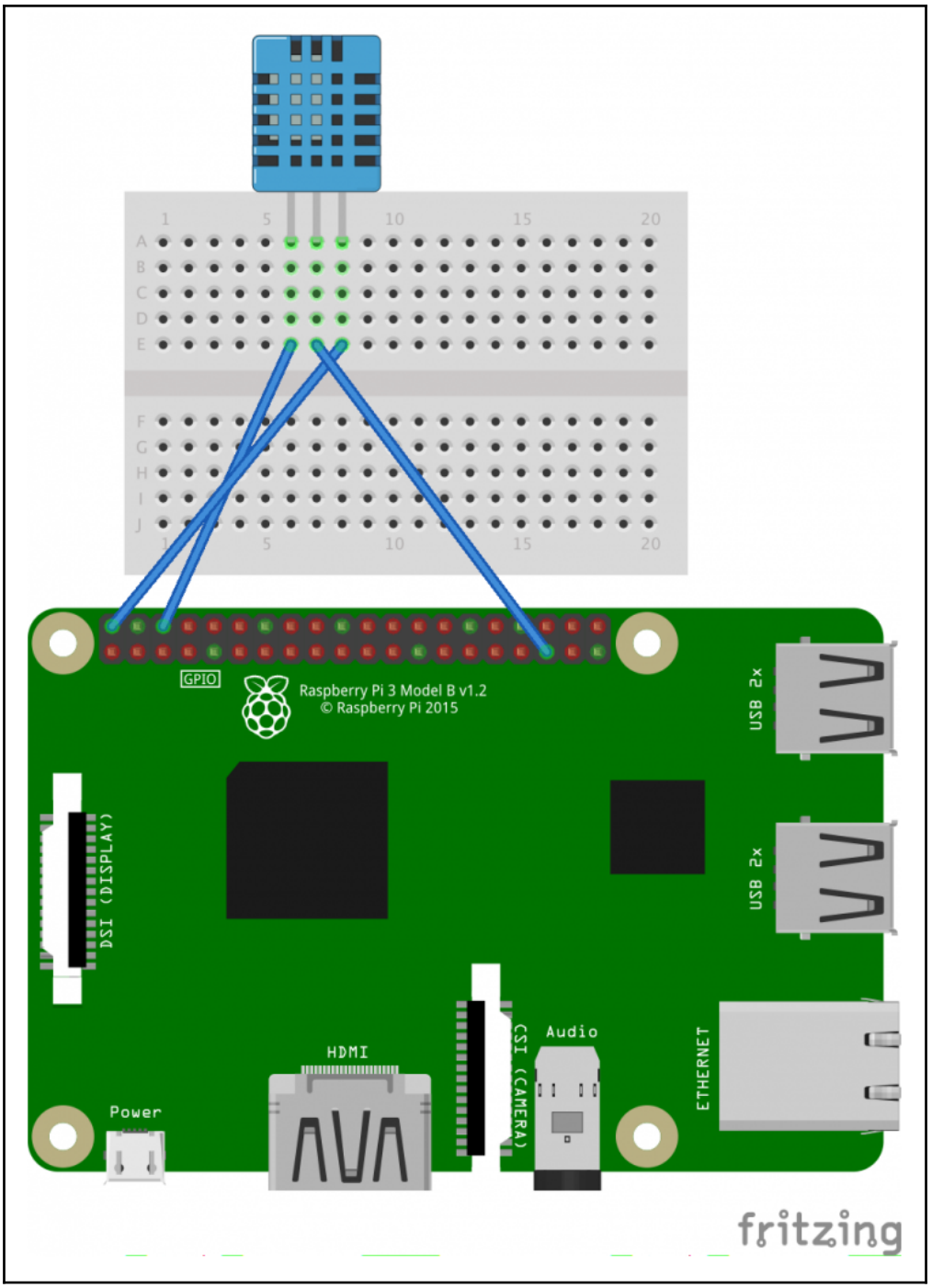

4.000
HC-SR04
VCC
Trig
Echo
GND
GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing

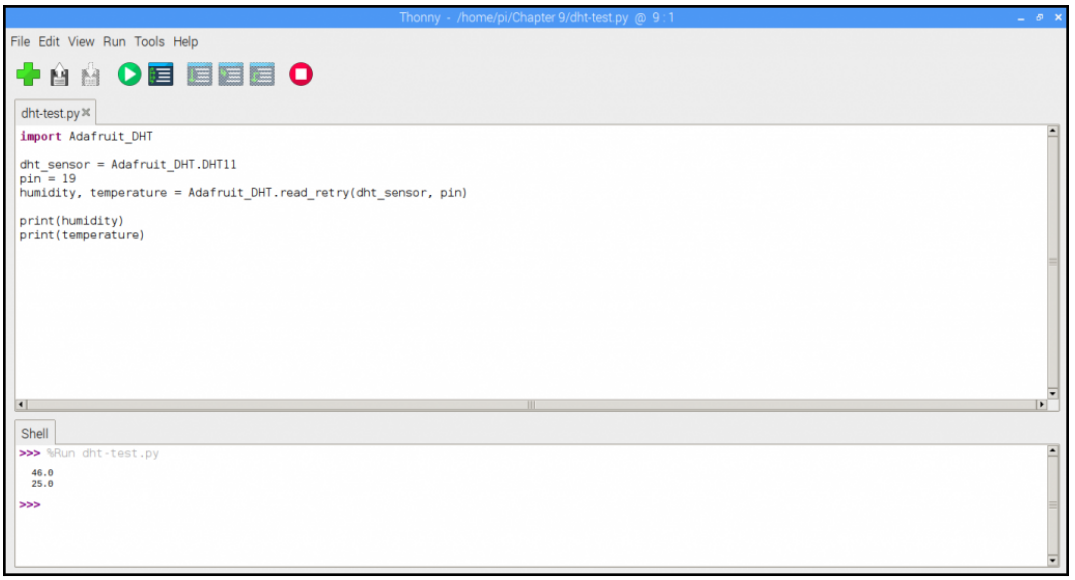
Chapter 18: Building a Home Security Dashboard

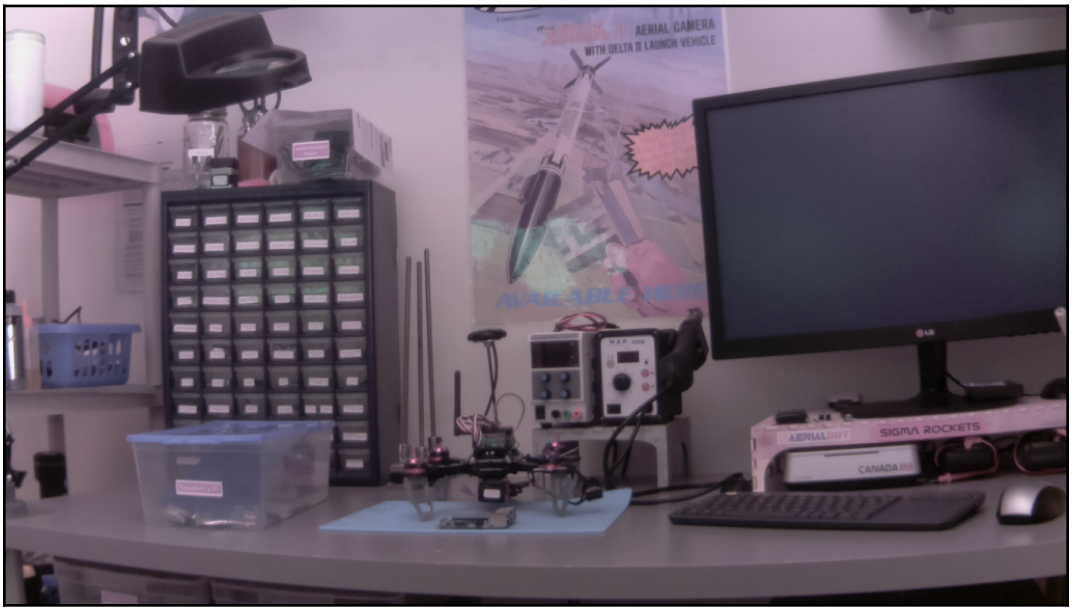


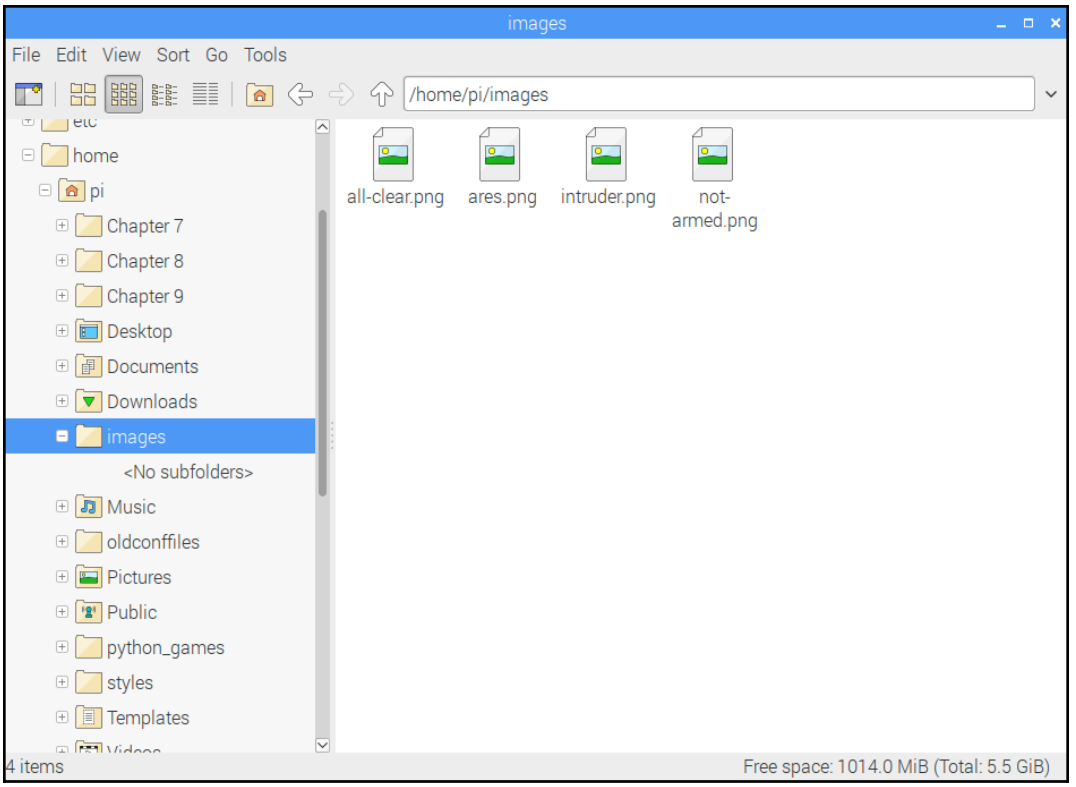

pi@raspberrypi: ~/Adafruit_Python_DHT
File Edit Tabs Help
pi@raspberrypi:~ $ cd Adafruit_Python_DHT/
pi@raspberrypi:~/Adafruit_Python_DHT $ ls
Adafruit_DHT build examples ez_setup.pyc README.md source
Adafruit_DHT.egg-info dist ez_setup.py LICENSE setup.py
pi@raspberrypi:~/Adafruit_Python_DHT $ sudo python3 setup.py install
running install
running bdist_egg
running egg_info
writing dependency_links to Adafruit_DHT.egg-info/dependency_links.txt
writing top-level names to Adafruit_DHT.egg-info/top_level.txt
writing Adafruit_DHT.egg-info/PKG-INFO
reading manifest file 'Adafruit_DHT.egg-info/SOURCES.txt'
writing manifest file 'Adafruit_DHT.egg-info/SOURCES.txt'
installing library code to build/bdist.linux-armv7l/egg
running install_lib
running build_py
creating build/lib.linux-armv7l-3.5
creating build/lib.linux-armv7l-3.5/Adafruit_DHT
copying Adafruit_DHT/common.py -> build/lib.linux-armv7l-3.5/Adafruit_DHT
copying Adafruit_DHT/Beaglebone_Black.py -> build/lib.linux-armv7l-3.5/Adafruit_DHT
copying Adafruit_DHT/platform_detect.py -> build/lib.linux-armv7l-3.5/Adafruit_DHT
copying Adafruit_DHT/Test.py -> build/lib.linux-armv7l-3.5/Adafruit_DHT

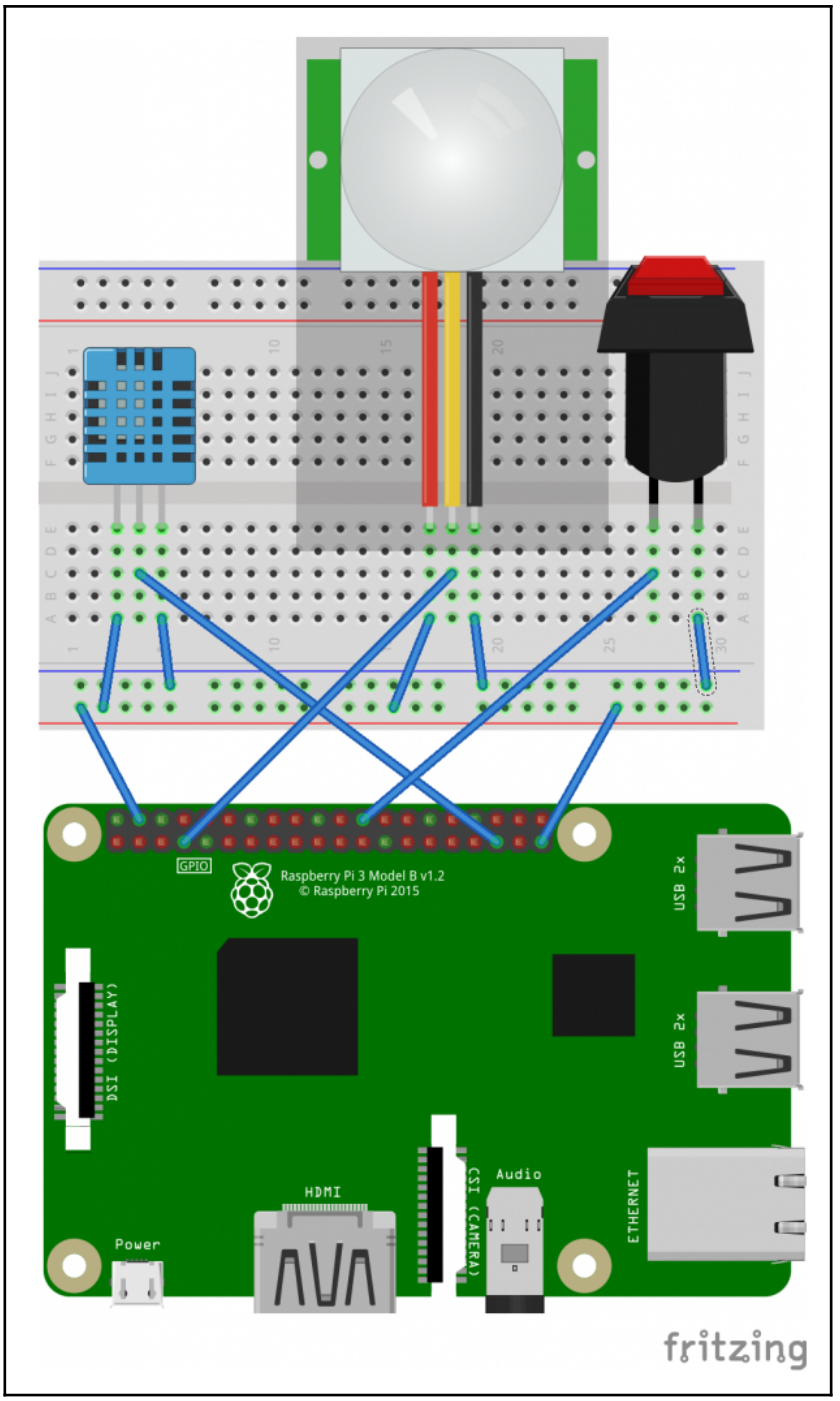

Manage packages for /usr/bin/python3
Search
<INSTALL>
Adafruit-DHT
automationhat
blinker
blinkt
buttonshim
Cap1xxx
chardet
click
colorama
cryptography
drumhat
envirophat
ExplorerHAT
Adafruit-DHT
Installed version: 1.3.2
Could not find the package from PyPI.
Close

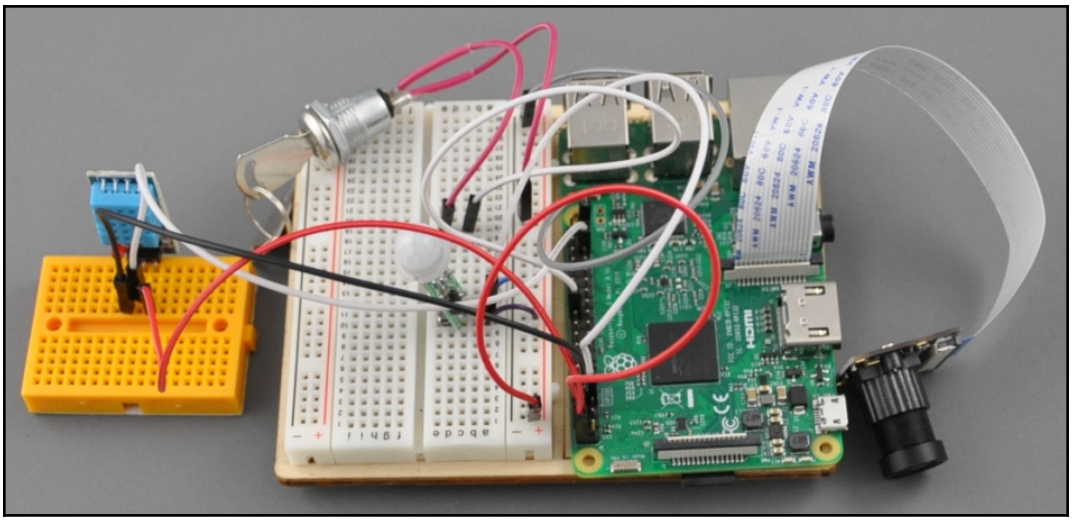

GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing


Thonny - /home/pi/Chapter 9/dht-test.py @ 9 : 1
File Edit View Run Tools Help
dht-test.py
import Adafruit_DHT
dht_sensor = Adafruit_DHT.DHT11
pin = 19
humidity, temperature = Adafruit_DHT.read_retry(dht_sensor, pin)
print(humidity)
print(temperature)
Shell
>>> %Run dht-test.py
46.0
25.0
>>>


AERIAL CAMERA
WITH DELTA II LAUNCH VEHICLE
SIGMA ROCKETS

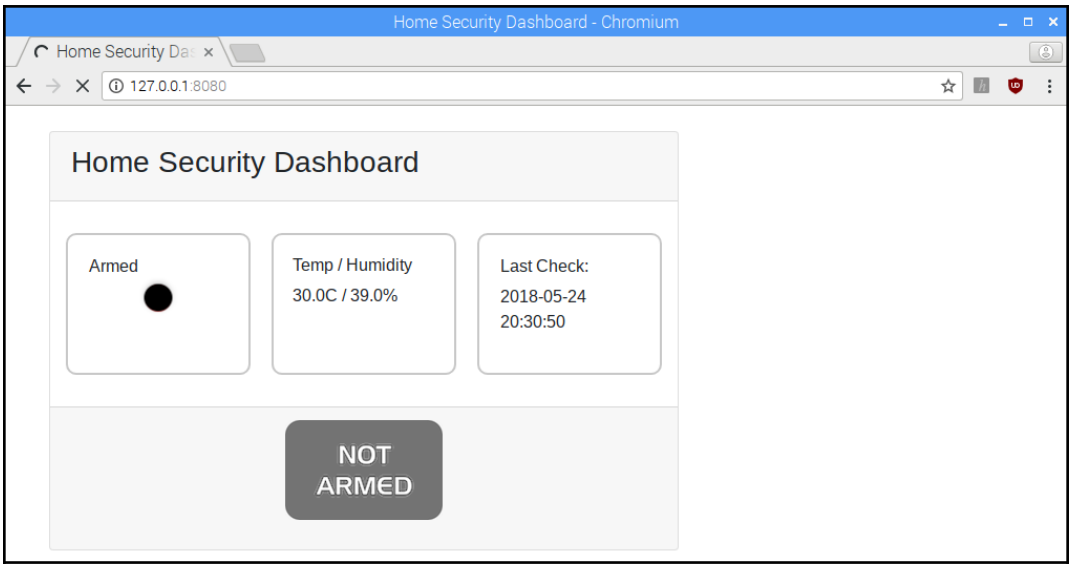

images
File Edit View Sort Go Tools
/home/pi/images
home
pi
Chapter 7
Chapter 8
Chapter 9
Desktop
Documents
Downloads
images
<No subfolders>
Music
oldconffiles
Pictures
Public
python_games
styles
Templates
all-clear.png
ares.png
intruder.png
not-armed.png
4 items
Free space: 1014.0 MiB (Total: 5.5 GiB)

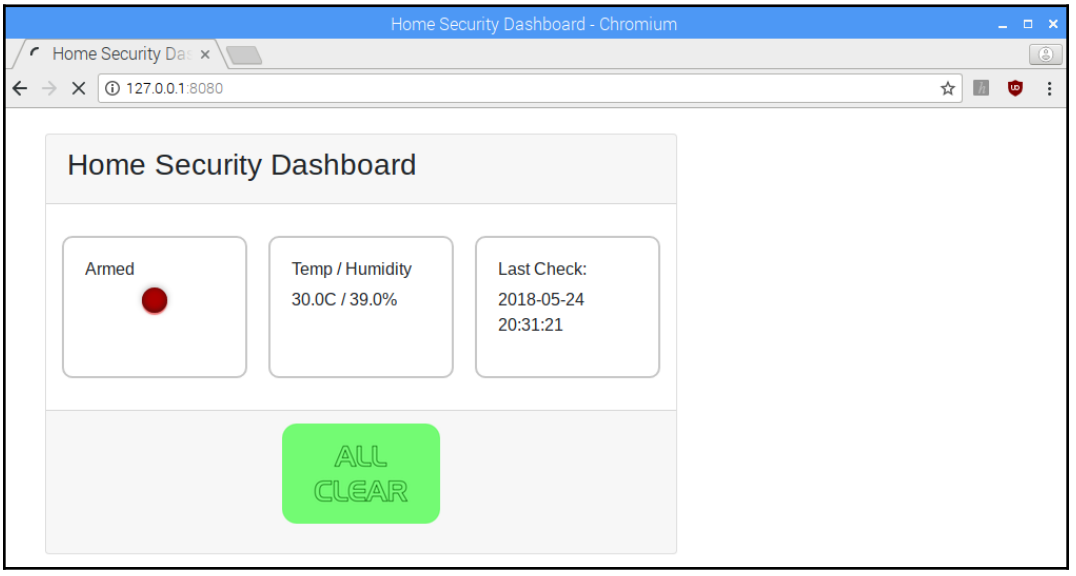

GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing

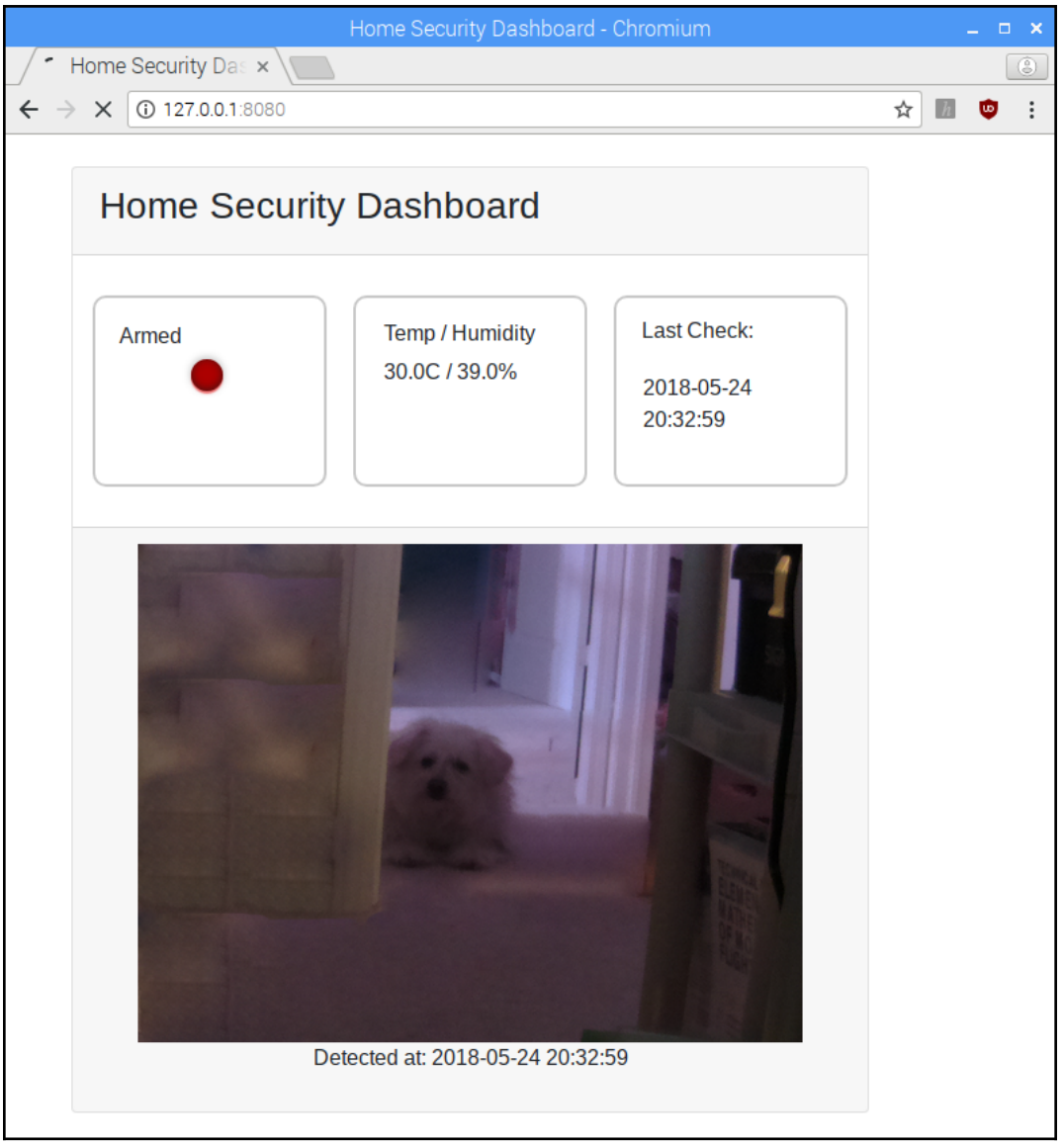


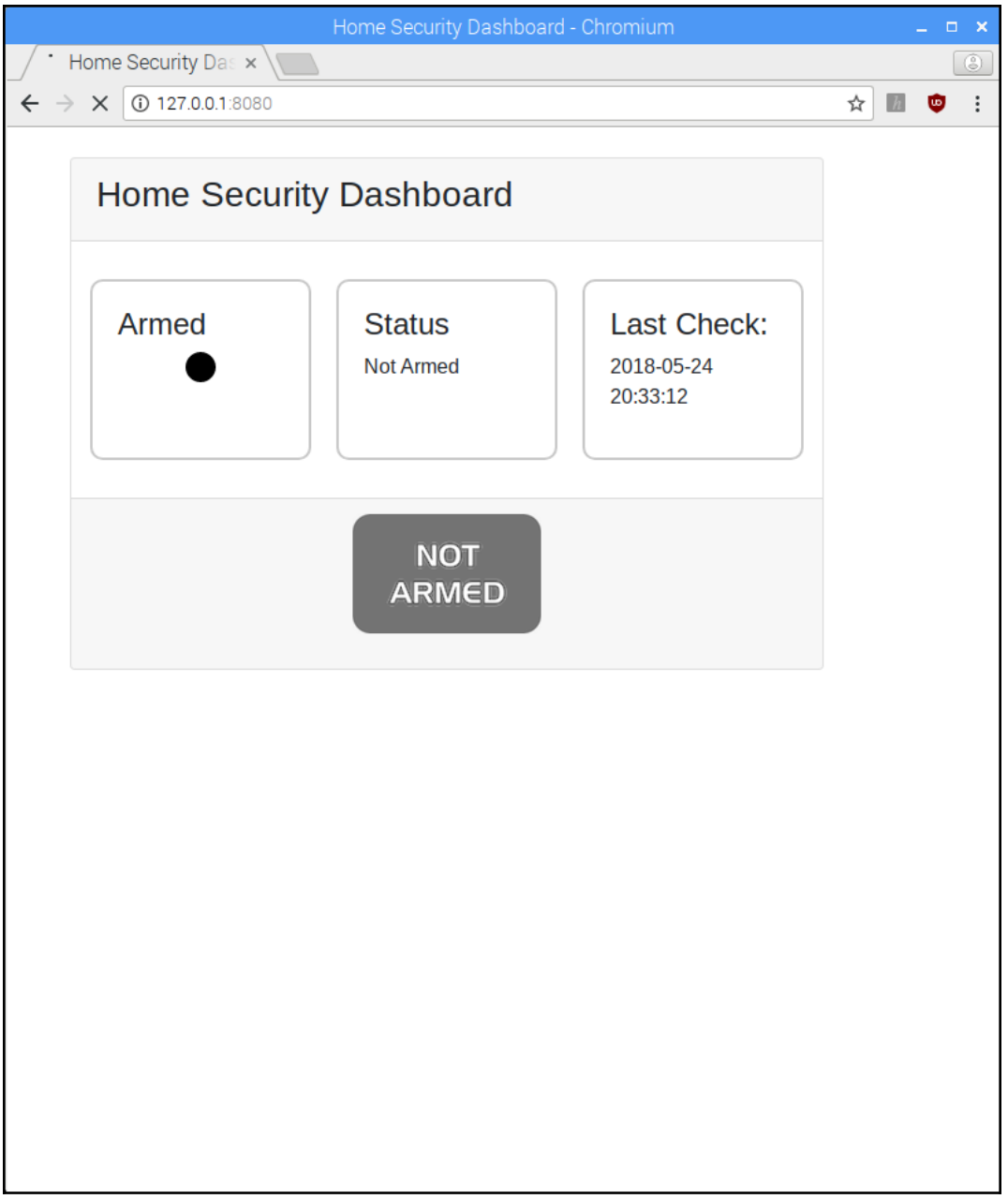

NOT
ARMED

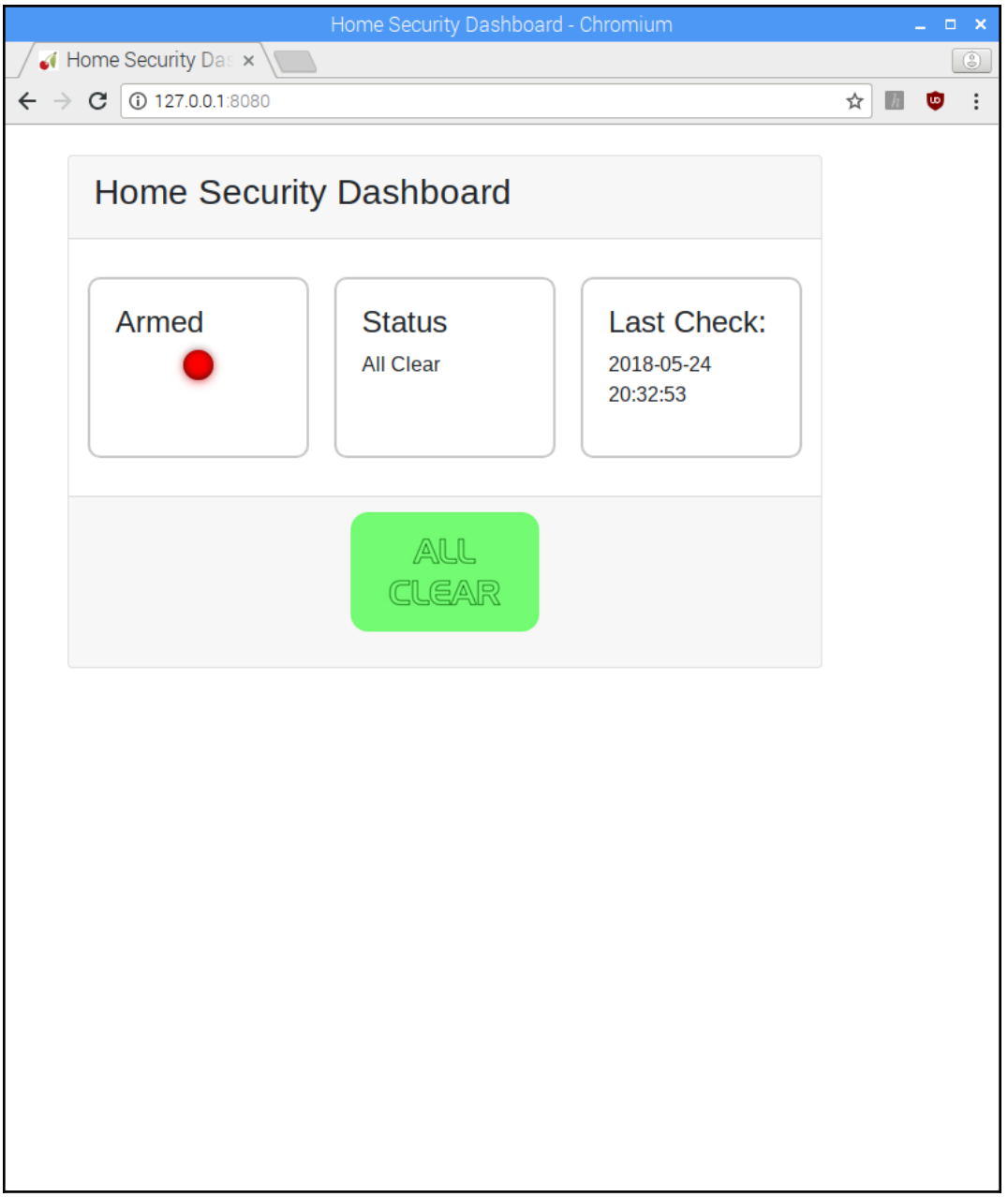

ALL
CLEAR

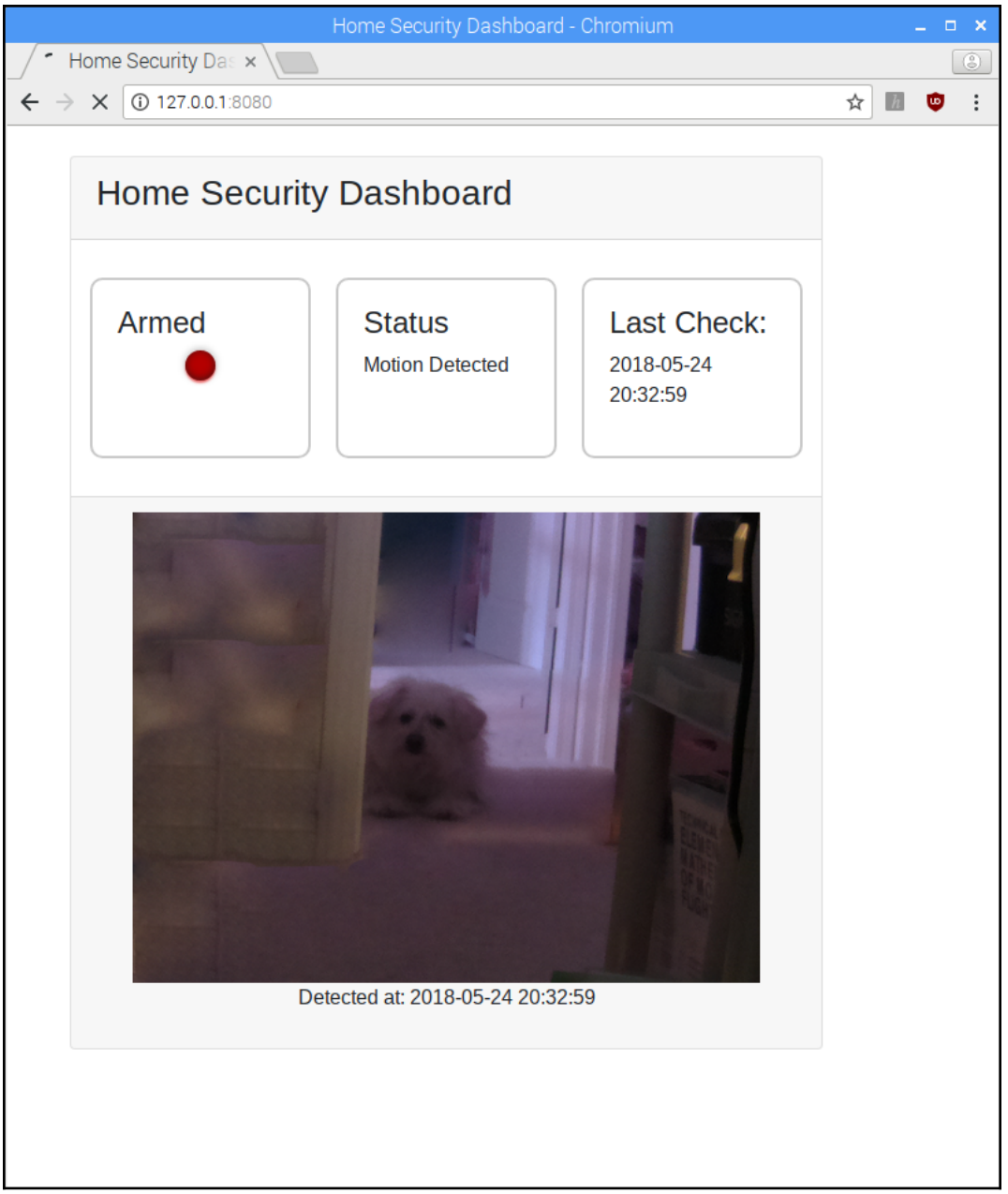

Home Security Dashboard - Chromium
Home Security Da
127.0.0.1:8080
Home Security Dashboard
Armed
Temp / Humidity
30.0C / 39.0%
Last Check:
2018-05-24
20:30:50
NOT ARMED


Home Security Dashboard - Chromium
Home Security Da
127.0.0.1:8080
Home Security Dashboard
Armed
Temp / Humidity
30.0C / 39.0%
Last Check:
2018-05-24
20:31:21
ALL CLEAR

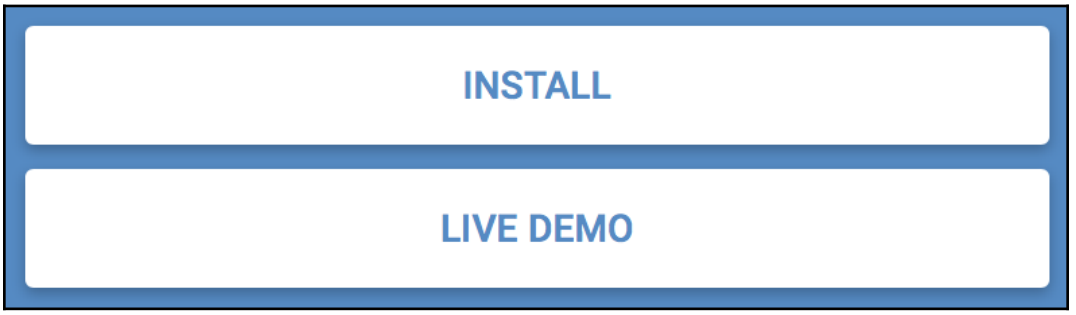

Home Security Dashboard - Chromium
Home Security Da
127.0.0.1:8080
Home Security Dashboard
Armed
Temp / Humidity
30.0C / 39.0%
Last Check:
2018-05-24
20:32:59
Detected at: 2018-05-24 20:32:59

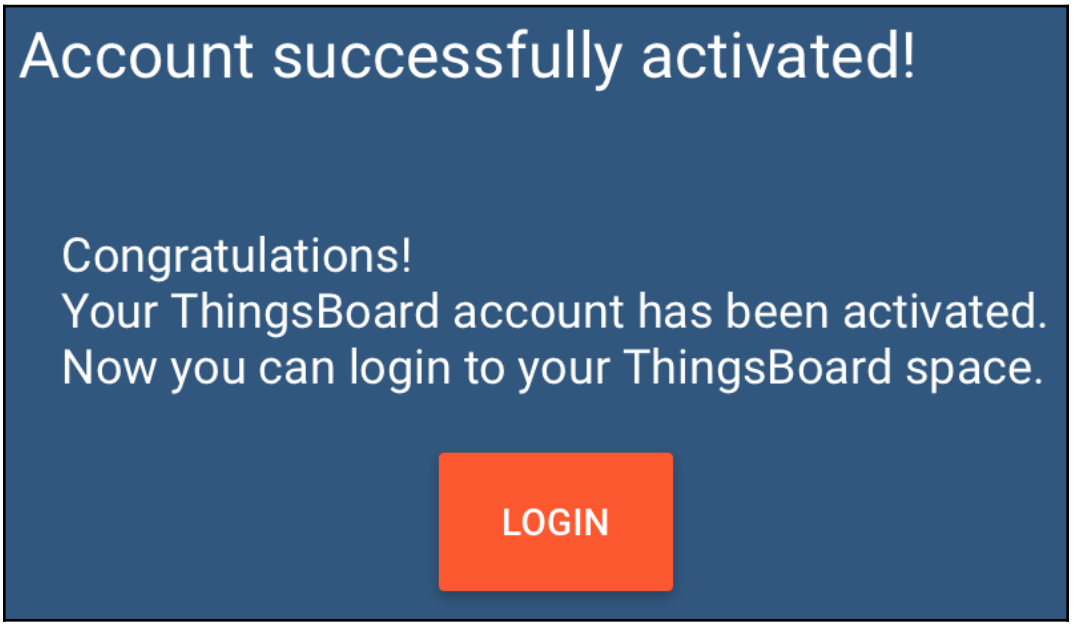

Home Security Dashboard - Chromium
Home Security Das
127.0.0.1:8080
Home Security Dashboard
Armed
Status
Not Armed
Last Check:
2018-05-24
20:33:12
NOT ARMED

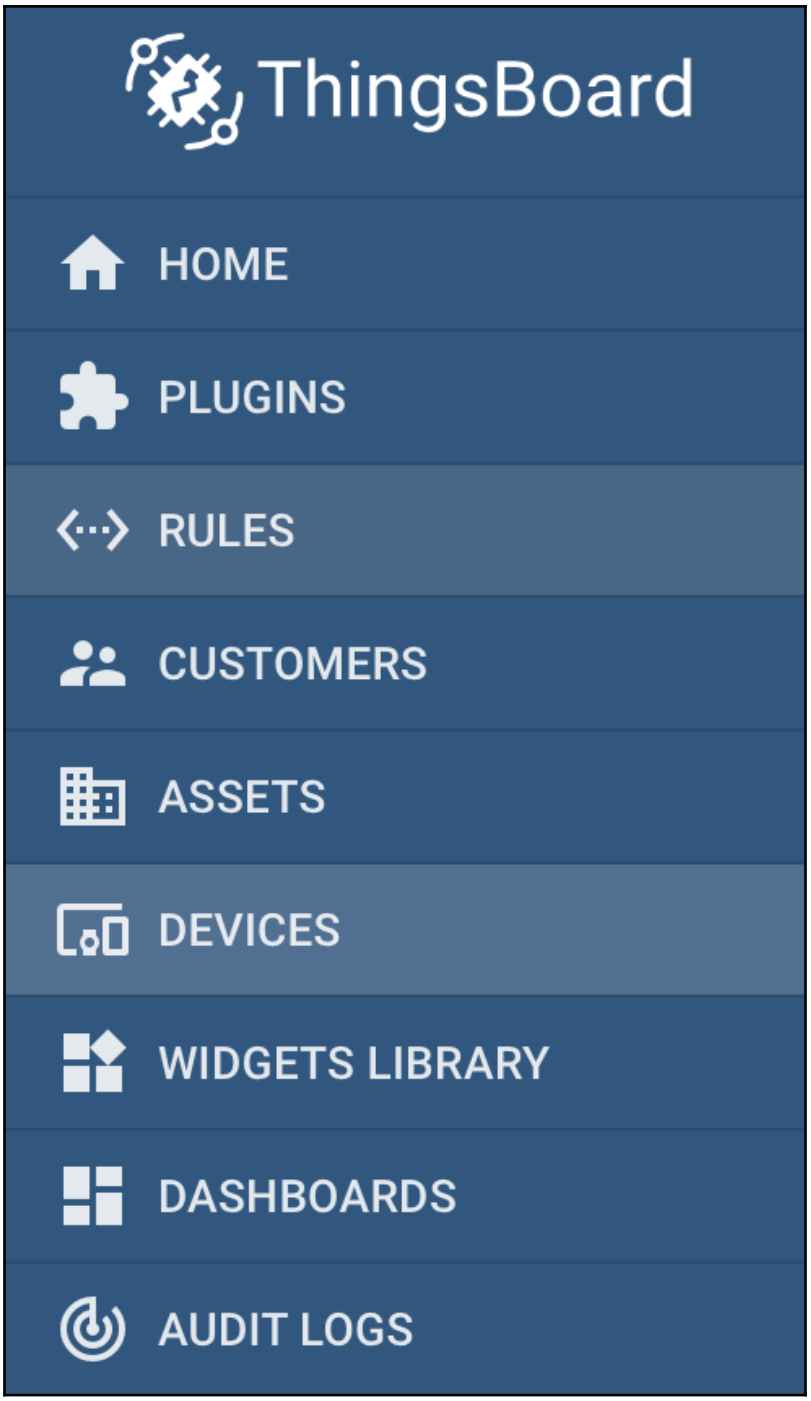

Home Security Dashboard - Chromium
Home Security Da
127.0.0.1:8080
Home Security Dashboard
Armed
Status
All Clear
Last Check:
2018-05-24
20:32:53
ALL
CLEAR

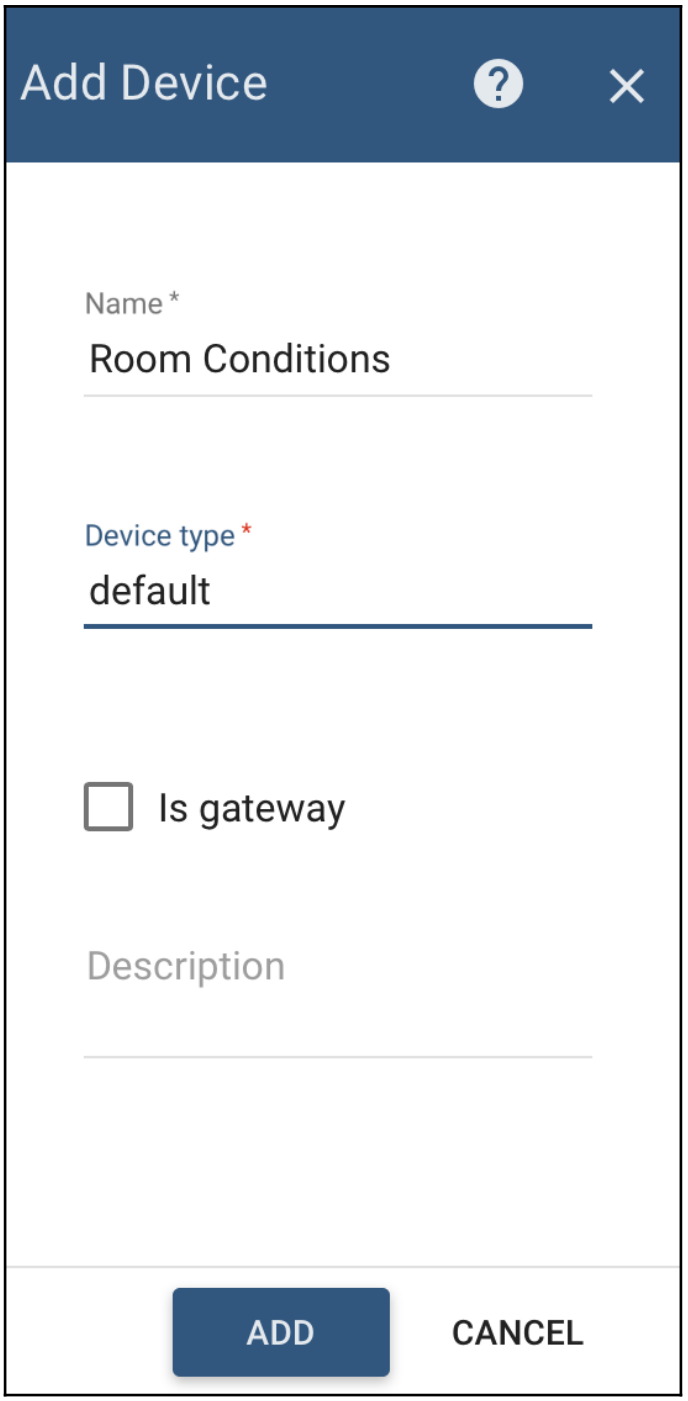

Home Security Dashboard - Chromium
Home Security Das
127.0.0.1:8080
Home Security Dashboard
Armed
Status
Motion Detected
Last Check:
2018-05-24
20:32:59
Detected at: 2018-05-24 20:32:59

Chapter 19: Publishing to Web Services



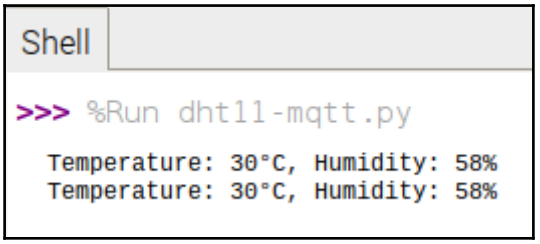


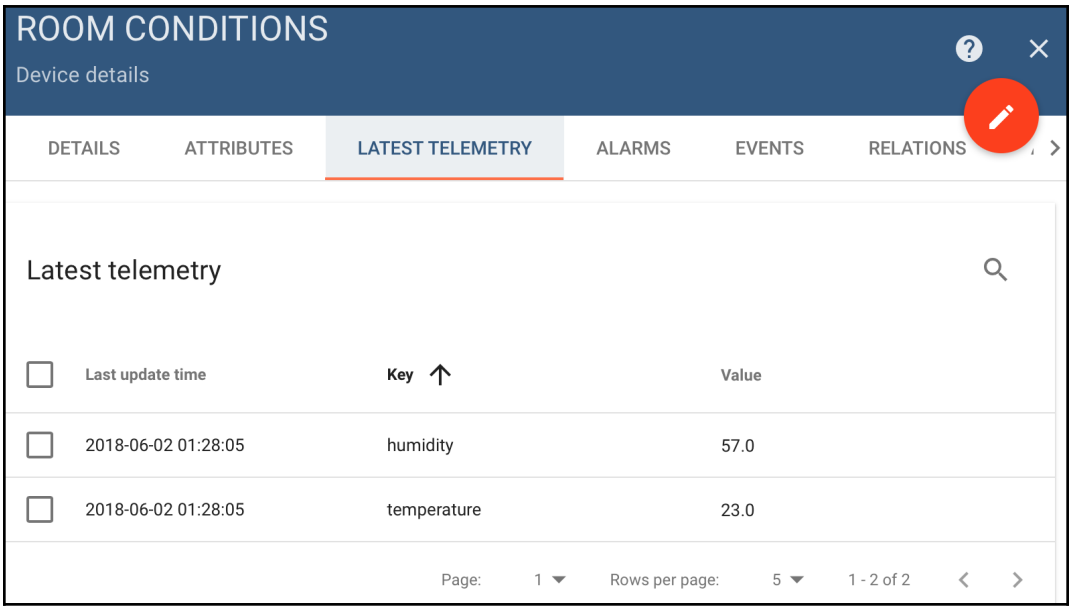

ThingsBoard
HOME
PLUGINS
RULES
CUSTOMERS
ASSETS
DEVICES
WIDGETS LIBRARY
DASHBOARDS
AUDIT LOGS

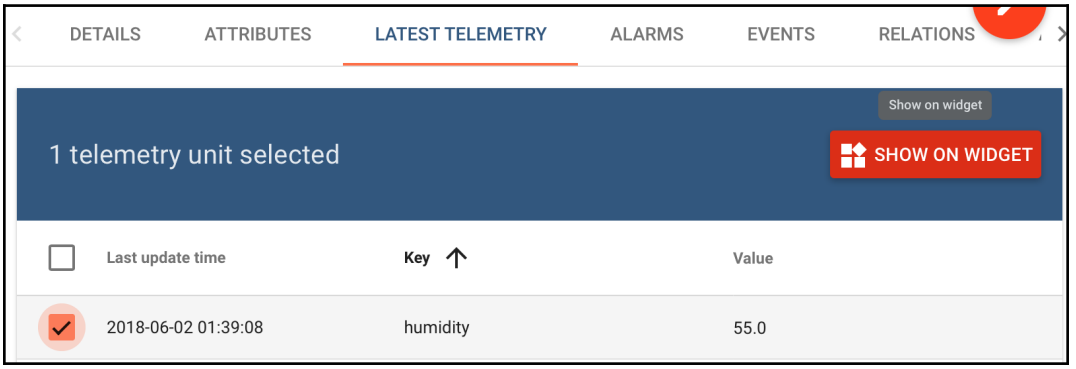

Add Device
Name *
Room Conditions
Device type *
default
Is gateway
Description
ADD
CANCEL

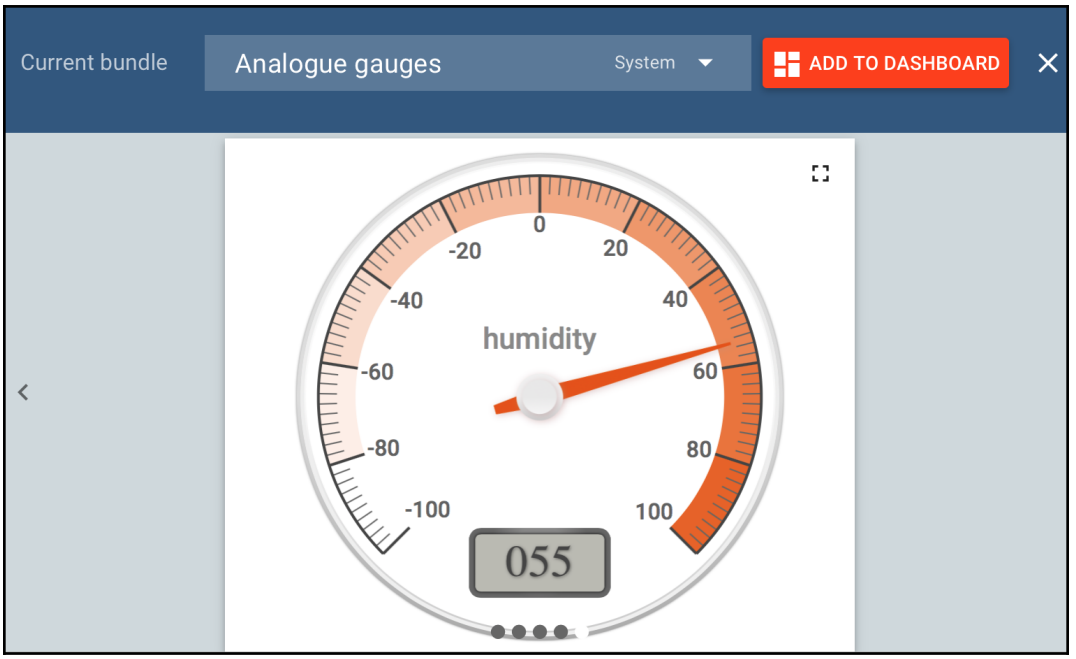

Room Conditions
DEFAULT

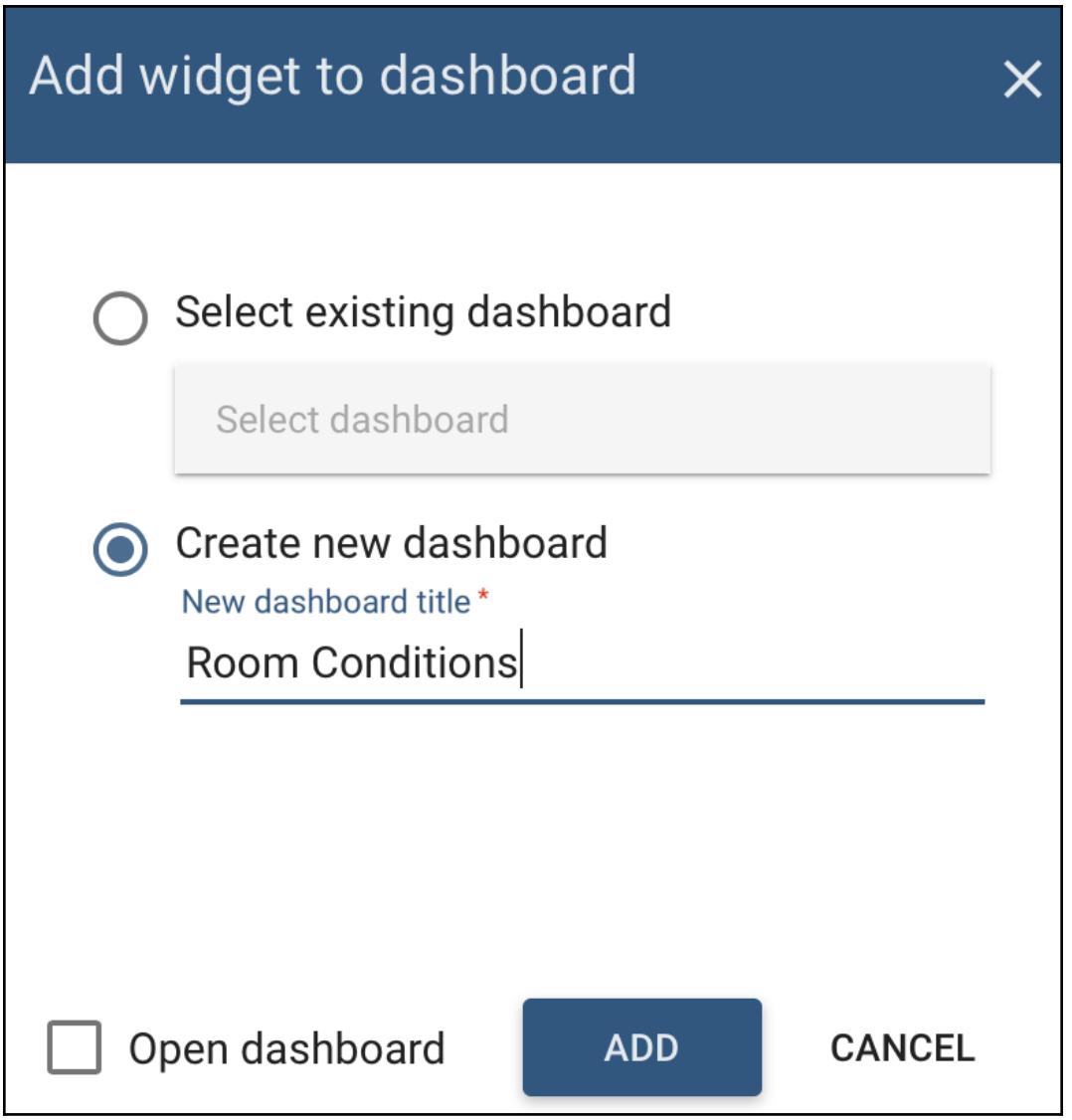

COPY ACCESS TOKEN

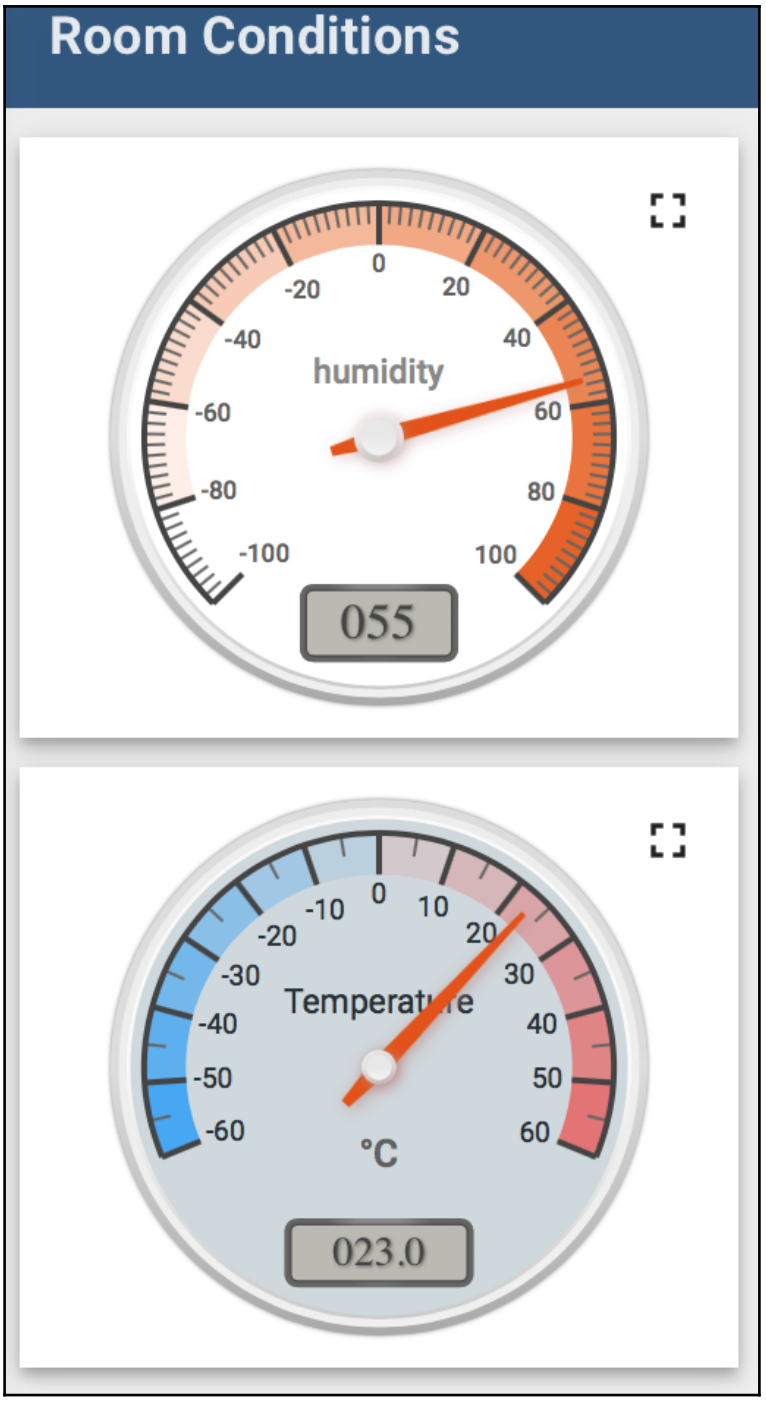

Shell
>>> %Run dht11-mqtt.py
Temperature: 30°C, Humidity: 58%
Temperature: 30°C, Humidity: 58%


ROOM CONDITIONS
Device details
DETAILS
ATTRIBUTES
LATEST TELEMETRY
ALARMS
EVENTS
RELATIONS
Latest telemetry
Last update time
Key
Value
2018-06-02 01:28:05
humidity
57.0
2018-06-02 01:28:05
temperature
23.0
Page:
1
Rows per page:
5
1 - 2 of 2

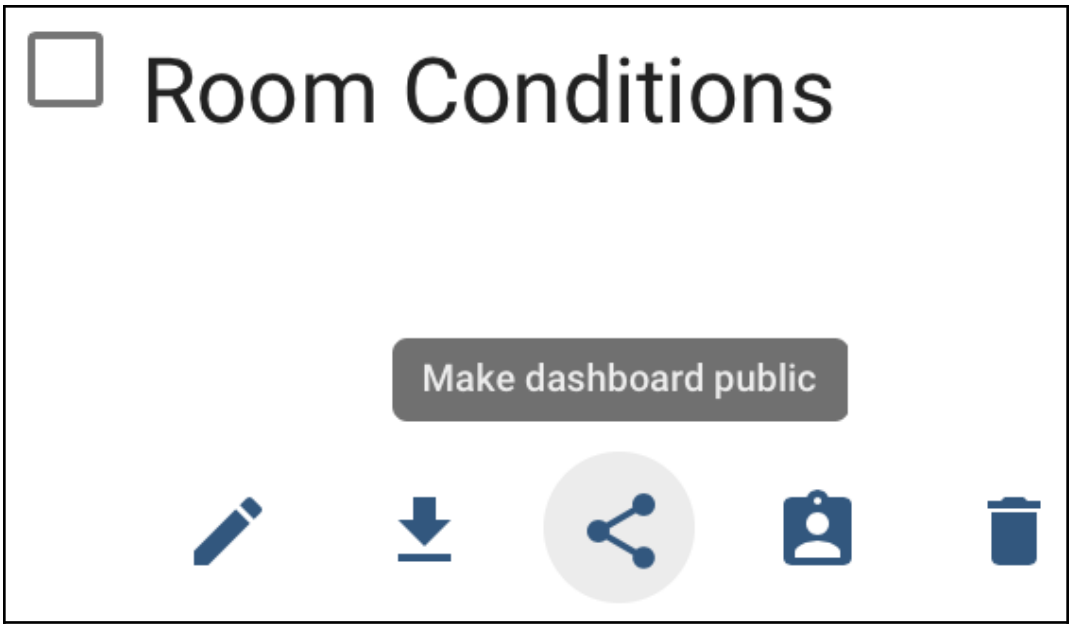

DETAILS
ATTRIBUTES
LATEST TELEMETRY
ALARMS
EVENTS
RELATIONS
1 telemetry unit selected
Show on widget
SHOW ON WIDGET
Last update time
Key
Value
2018-06-02 01:39:08
humidity
55.0


Current bundle
Analogue gauges
System
ADD TO DASHBOARD
0
-20
20
-40
40
humidity
-60
60
-80
80
-100
100
055

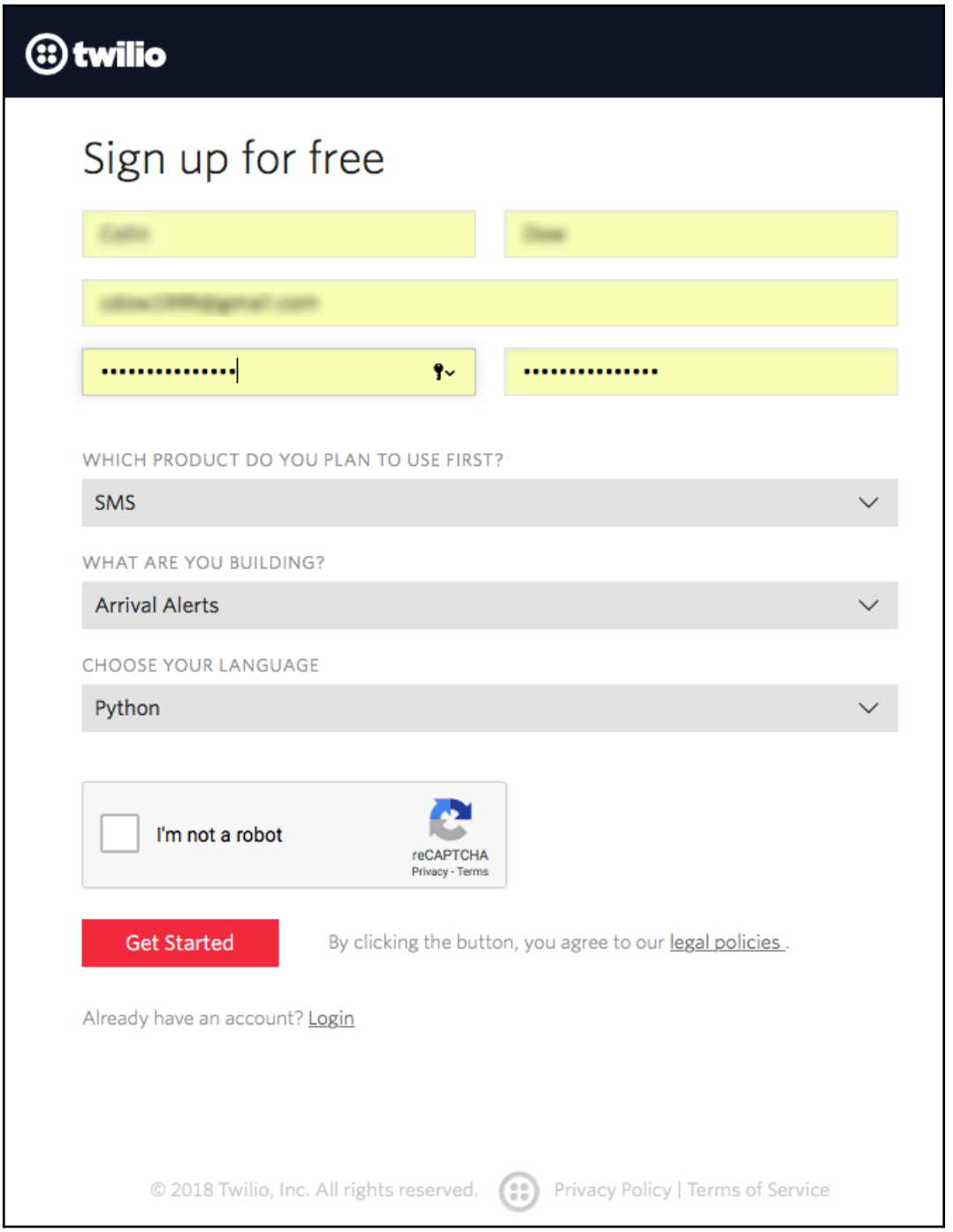

Add widget to dashboard
Select existing dashboard
Select dashboard
Create new dashboard
New dashboard title *
Room Conditions
Open dashboard
ADD
CANCEL

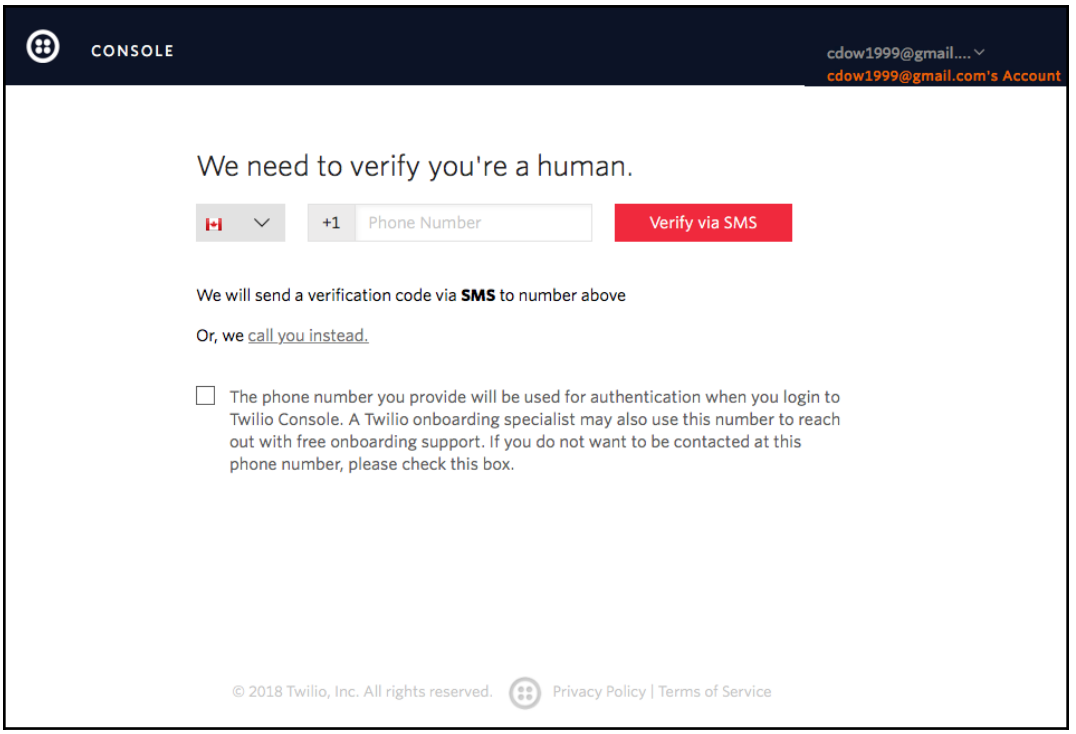

Room Conditions
0
-20
20
-40
40
humidity
-60
60
-80
80
-100
100
055
-10
10
-30
30
-40
40
-50
50
-60
60
°C
023.0

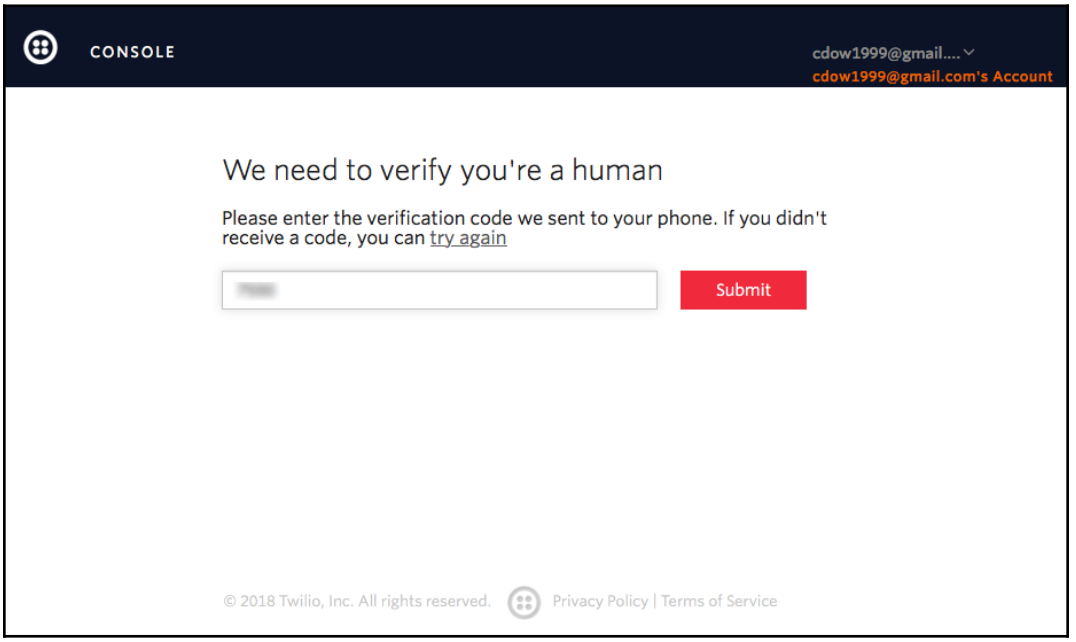

DASHBOARDS

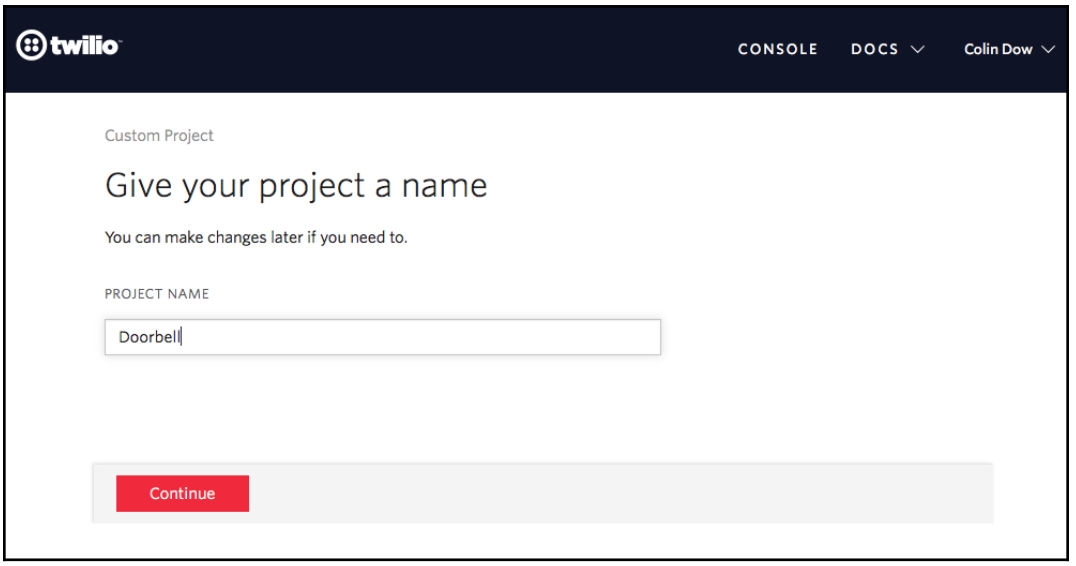

Room Conditions
Make dashboard public

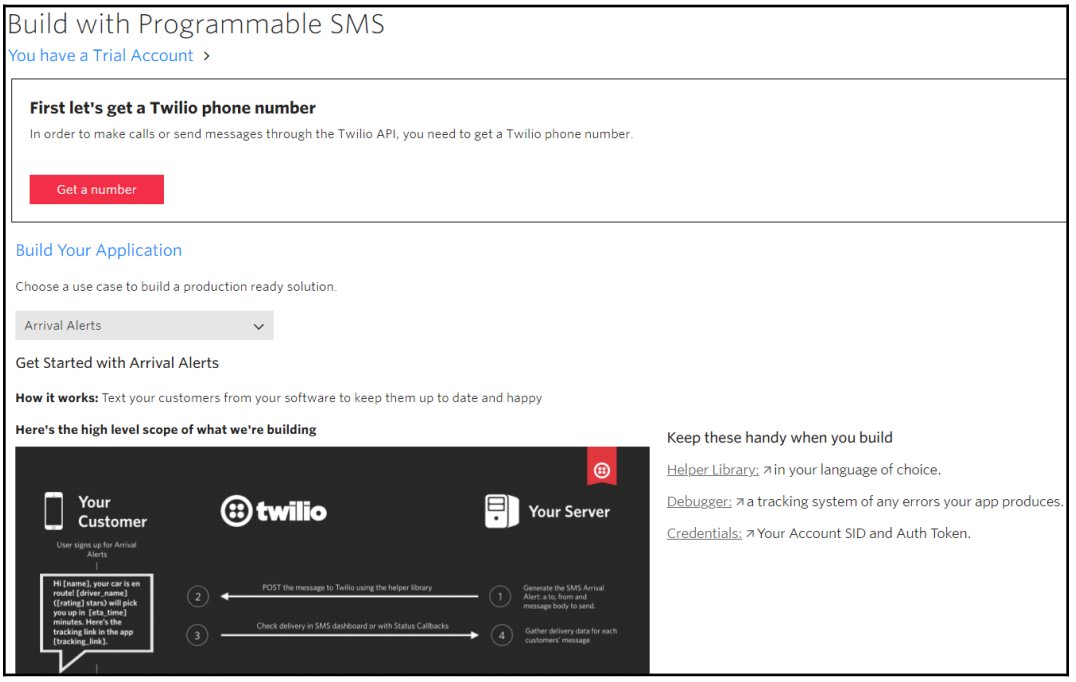

Dashboard is now public
Your dashboard Room Conditions is now public and accessible via next public link:
https://demo.thingsboard.io/dashboards/a552cf00-6629-11e8-bee0-c3b186e30863?publicId=674b91
Note: Do not forget to make related devices public in order to access their data.
OK


twilio
Sign up for free
WHICH PRODUCT DO YOU PLAN TO USE FIRST?
SMS
WHAT ARE YOU BUILDING?
Arrival Alerts
CHOOSE YOUR LANGUAGE
Python
I'm not a robot
reCAPTCHA
Privacy - Terms
Get Started
By clicking the button, you agree to our legal policies .
Already have an account? Login
© 2018 Twilio, Inc. All rights reserved.
Privacy Policy | Terms of Service

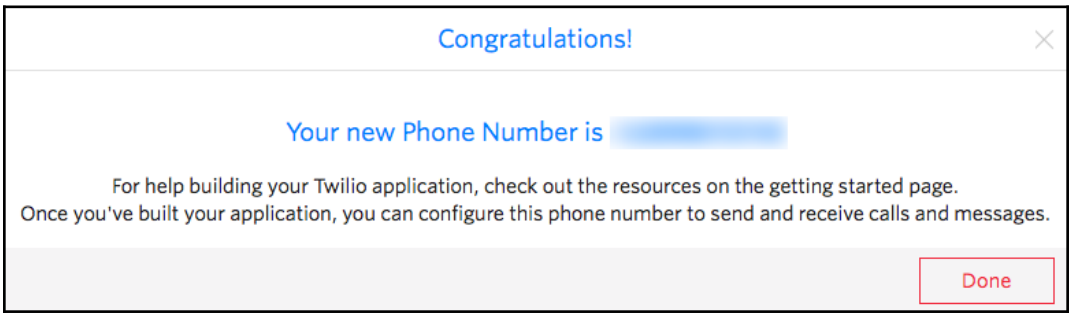

CONSOLE
cdow1999@gmail....
cdow1999@gmail.com's Account
We need to verify you're a human.
+1
Phone Number
Verify via SMS
We will send a verification code via **SMS** to number above
Or, we call you instead.
The phone number you provide will be used for authentication when you login to Twilio Console. A Twilio onboarding specialist may also use this number to reach out with free onboarding support. If you do not want to be contacted at this phone number, please check this box.
© 2018 Twilio, Inc. All rights reserved.
Privacy Policy | Terms of Service

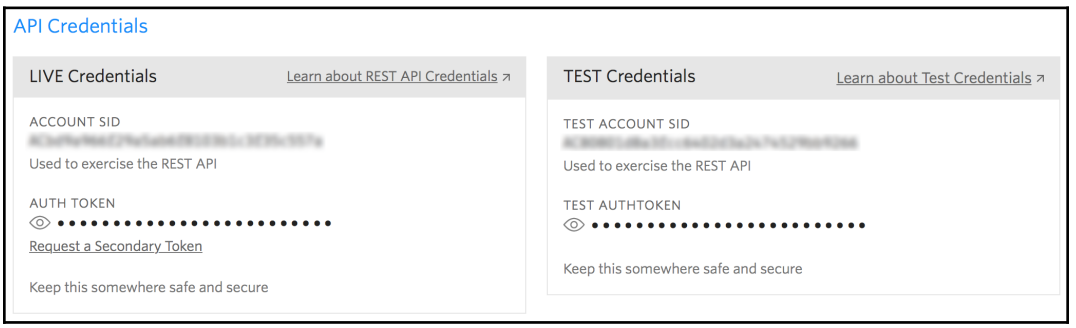

CONSOLE
cdow1999@gmail....
cdow1999@gmail.com's Account
We need to verify you're a human
Please enter the verification code we sent to your phone. If you didn't receive a code, you can try again
Submit
© 2018 Twilio, Inc. All rights reserved.
Privacy Policy | Terms of Service

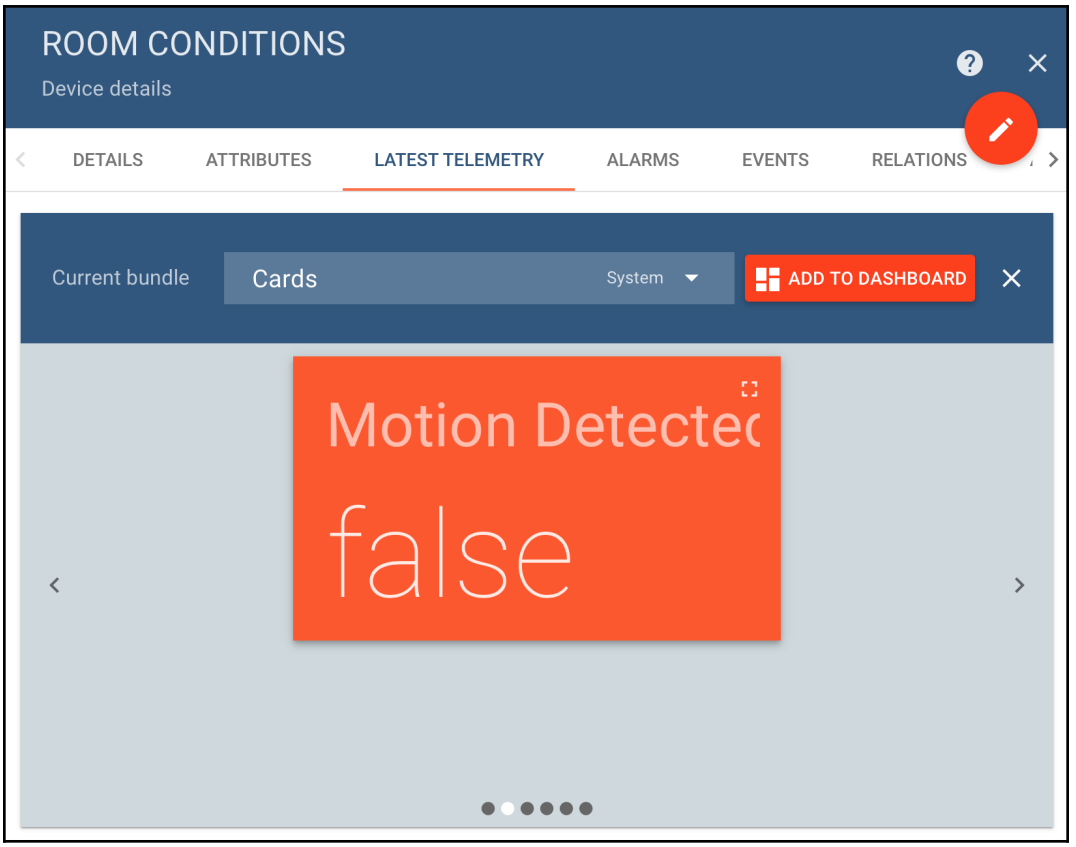

twilio
CONSOLE
DOCS
Colin Dow
Custom Project
Give your project a name
You can make changes later if you need to.
PROJECT NAME
Doorbell
Continue

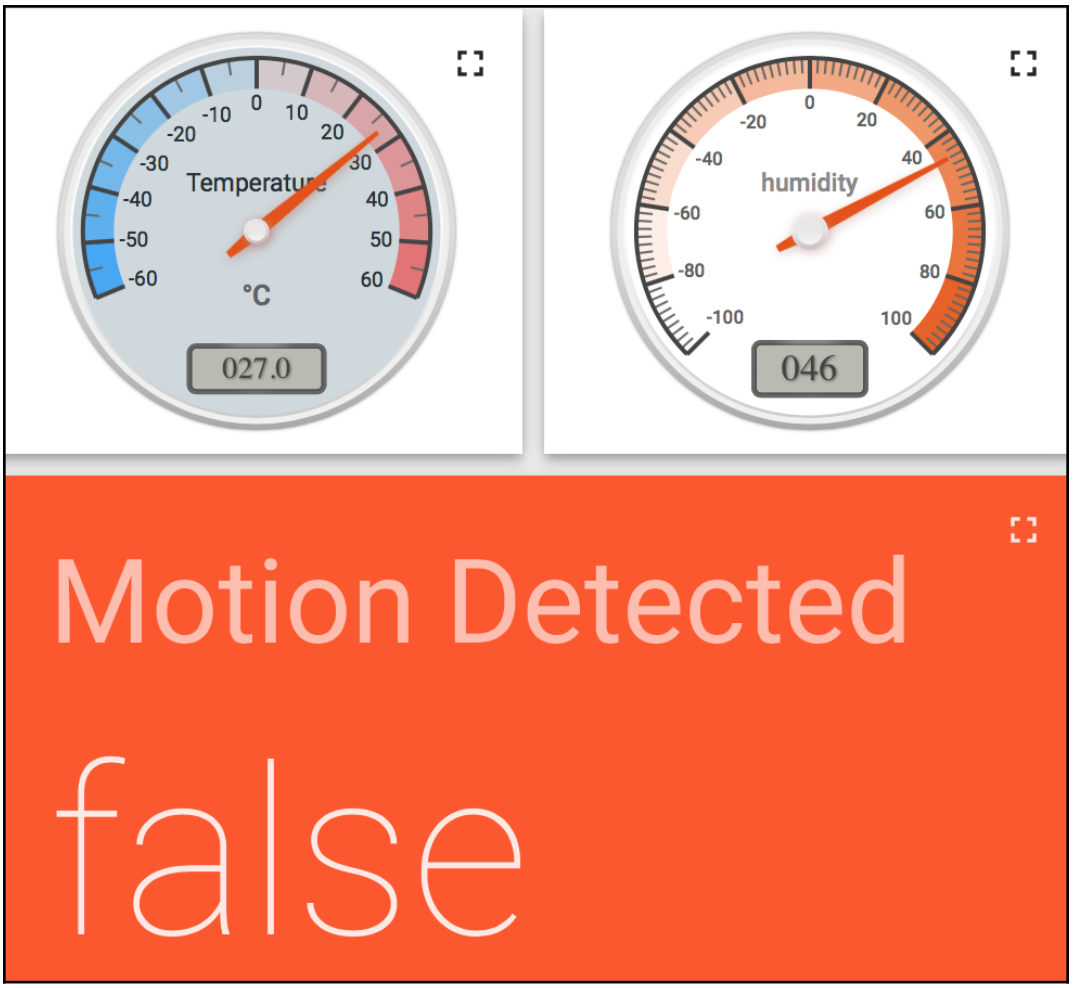

Build with Programmable SMS
You have a Trial Account
First let's get a Twilio phone number
In order to make calls or send messages through the Twilio API, you need to get a Twilio phone number.
Get a number
Build Your Application
Choose a use case to build a production ready solution.
Arrival Alerts
Get Started with Arrival Alerts
How it works: Text your customers from your software to keep them up to date and happy
Here's the high level scope of what we're building
Your Customer
User signs up for Arrival Alerts
twilio
Your Server
Hi {name}, your car is en route! {driver_name} ({rating} stars) will pick you up in {eta_time} minutes. Here's the tracking link in the app {tracking_link}.
2
POST the message to Twilio using the helper library
1
Generate the SMS Arrival Alert: a to, from and message body to send.
3
Check delivery in SMS dashboard or with Status Callbacks
4
Gather delivery data for each customers' message
Keep these handy when you build
Helper Library: in your language of choice.
Debugger: a tracking system of any errors your app produces.
Credentials: Your Account SID and Auth Token.


Your first Twilio Phone Number
Don't like this one? Search for a different number
This Canada phone number has the following capabilities:
Voice: This number can receive incoming calls and make outgoing calls.
SMS: This number can send and receive text messages to and from mobile numbers.
MMS: This number can send and receive multi media messages to and from mobile numbers.
Cancel
Choose this Number

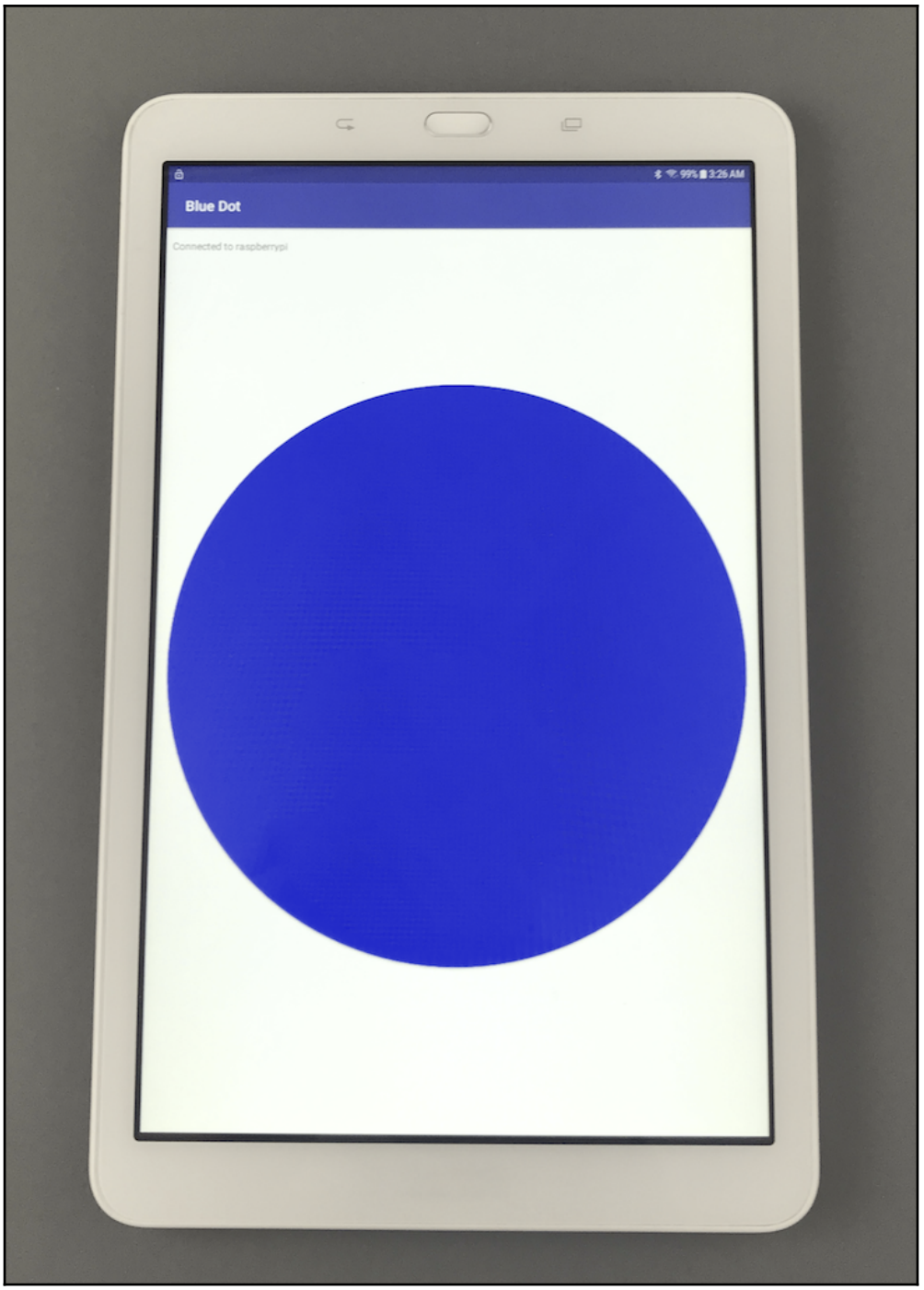

Congratulations!
Your new Phone Number is
For help building your Twilio application, check out the resources on the getting started page.
Once you've built your application, you can configure this phone number to send and receive calls and messages.
Done

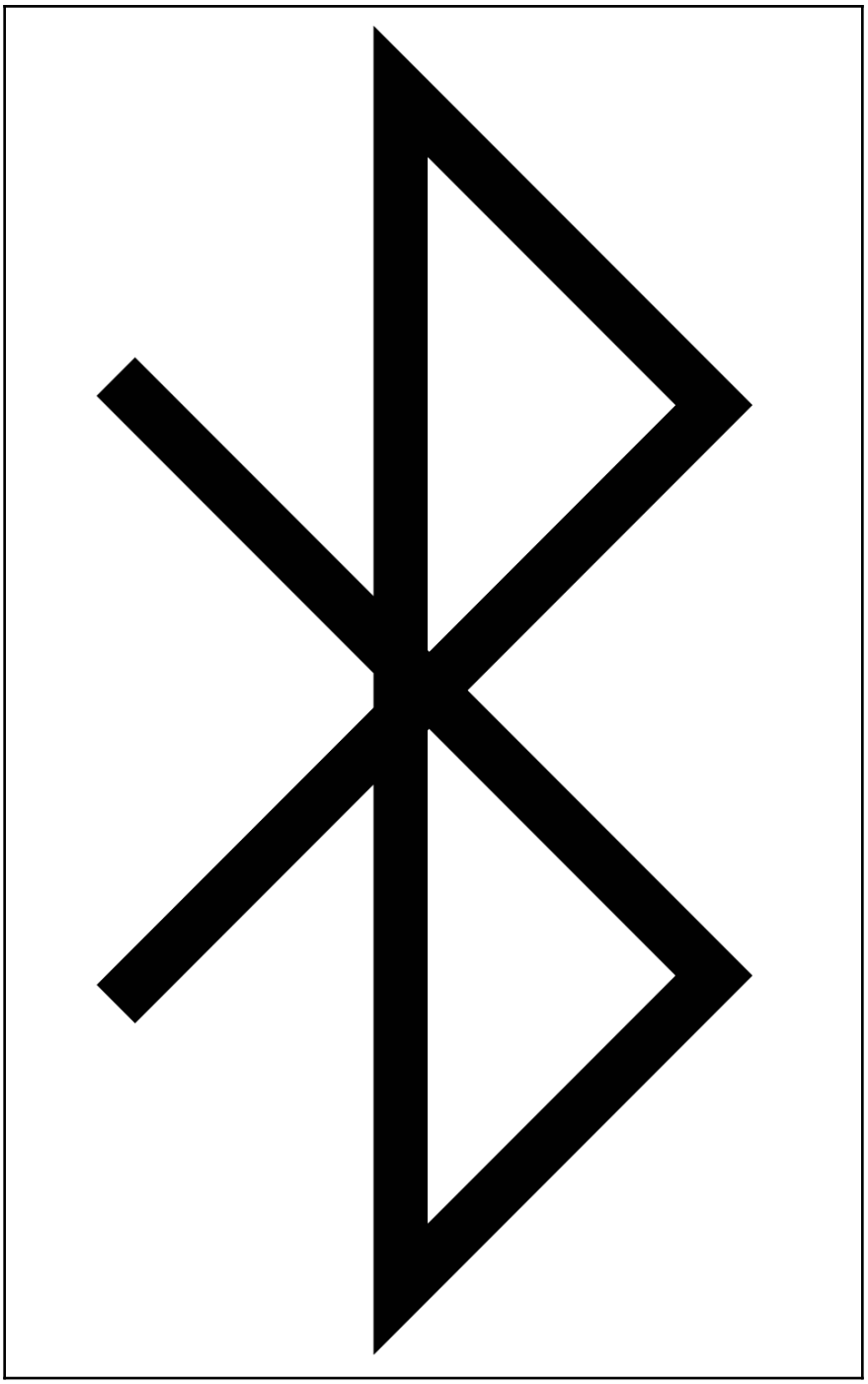

API Credentials
LIVE Credentials
Learn about REST API Credentials
ACCOUNT SID
Used to exercise the REST API
AUTH TOKEN
Request a Secondary Token
Keep this somewhere safe and secure
TEST Credentials
Learn about Test Credentials
TEST ACCOUNT SID
Used to exercise the REST API
TEST AUTHTOKEN
Keep this somewhere safe and secure

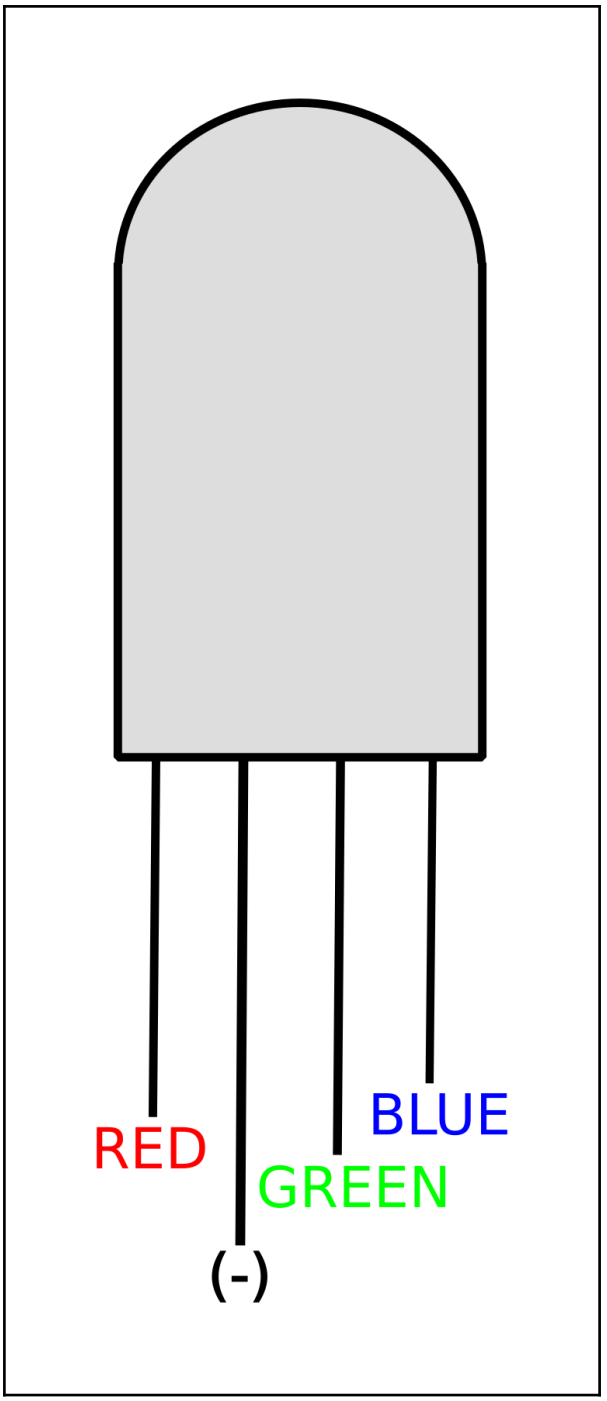

ROOM CONDITIONS
Device details
DETAILS
ATTRIBUTES
LATEST TELEMETRY
ALARMS
EVENTS
RELATIONS
Current bundle
Cards
System
ADD TO DASHBOARD
Motion Detectec
false

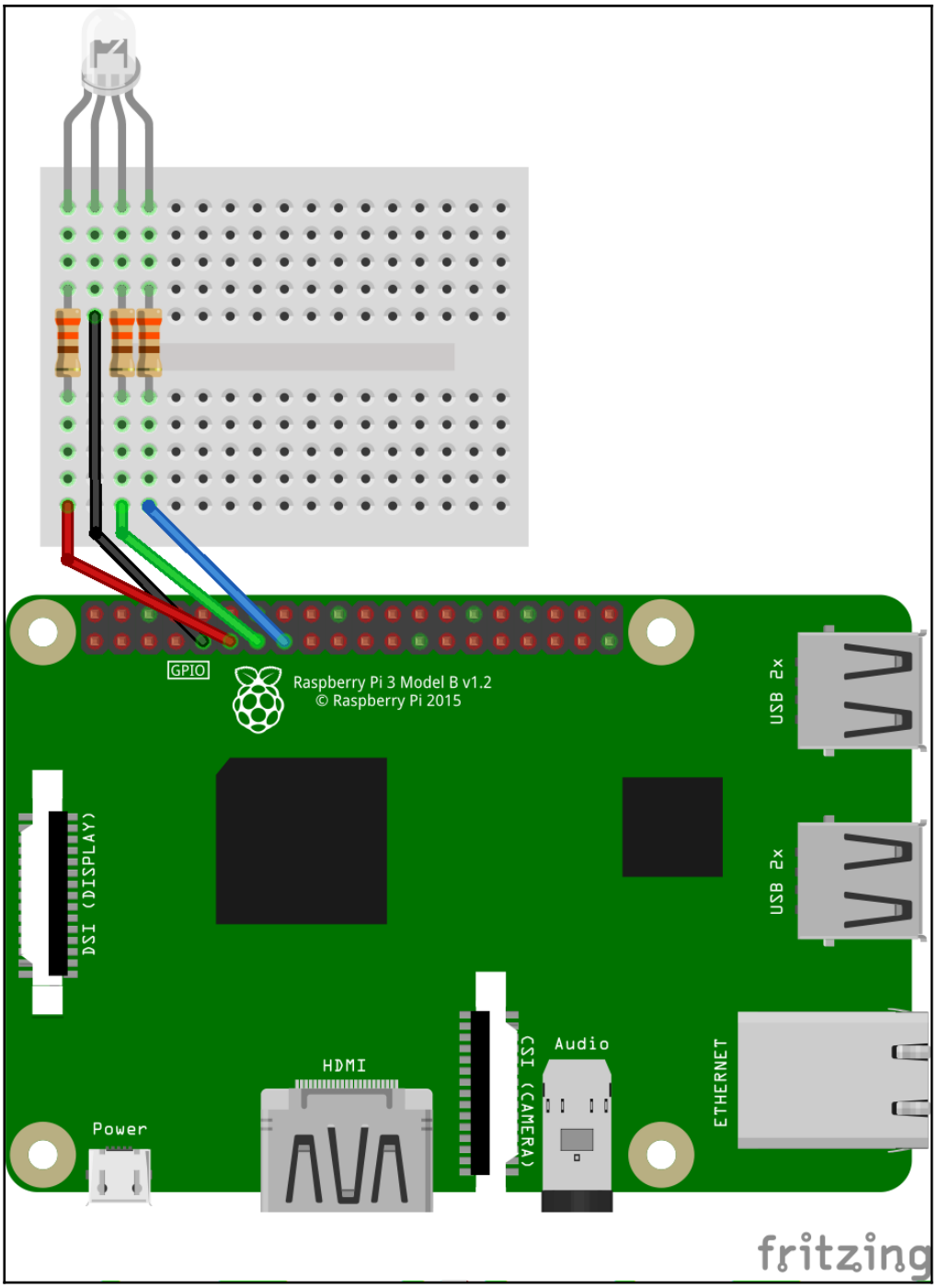

Temperature
°C
027.0
humidity
046
Motion Detected
false

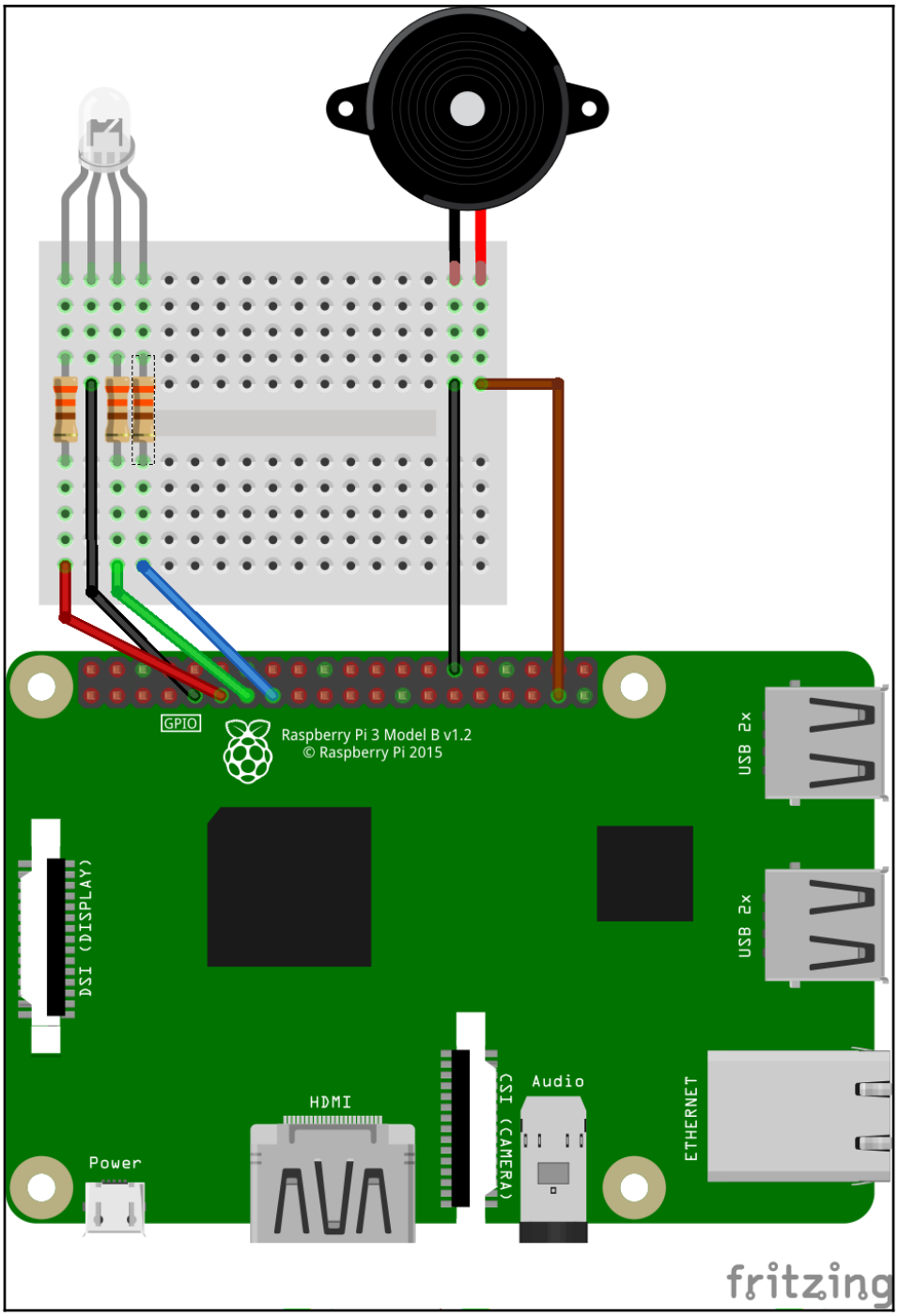
Chapter 20: Creating a Doorbell Button Using Bluetooth


99% 3:25 AM
Blue Dot
Connected to raspberrypi

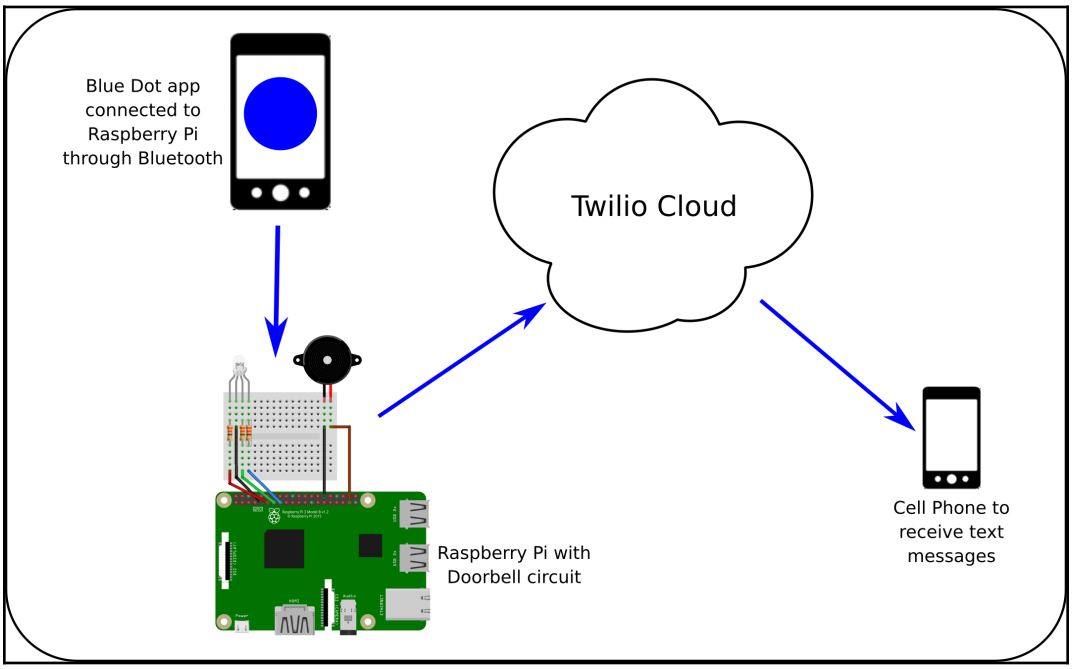



RED
(-)
GREEN
BLUE


GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing


GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power
fritzing

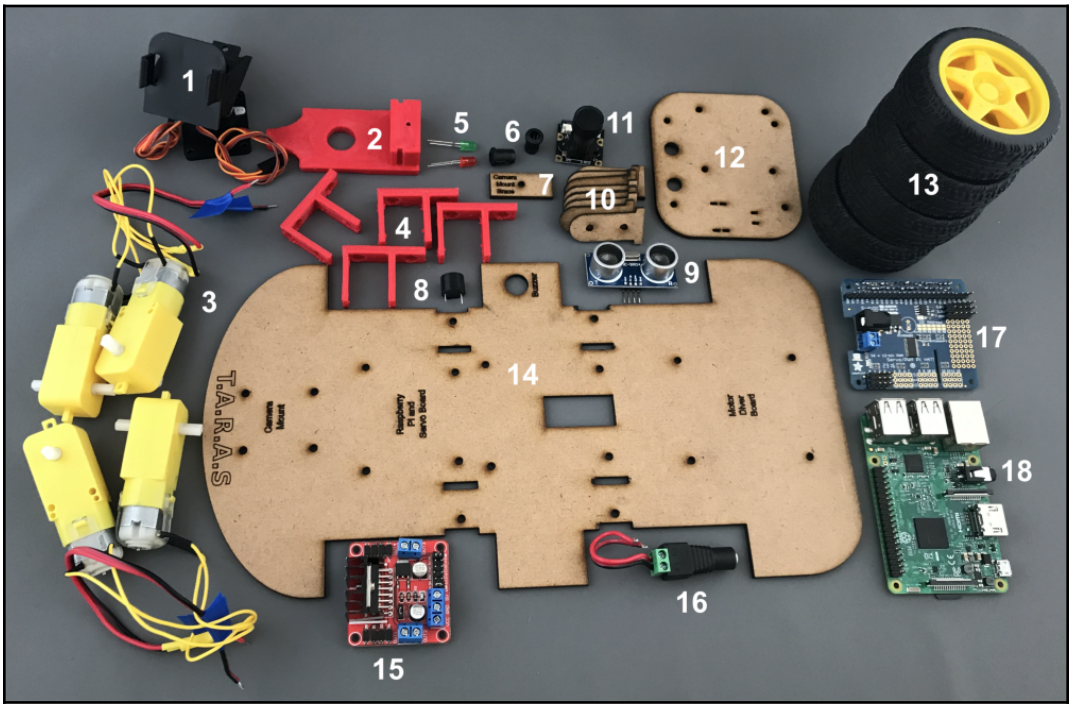
Chapter 21: Enhancing Our IoT Doorbell

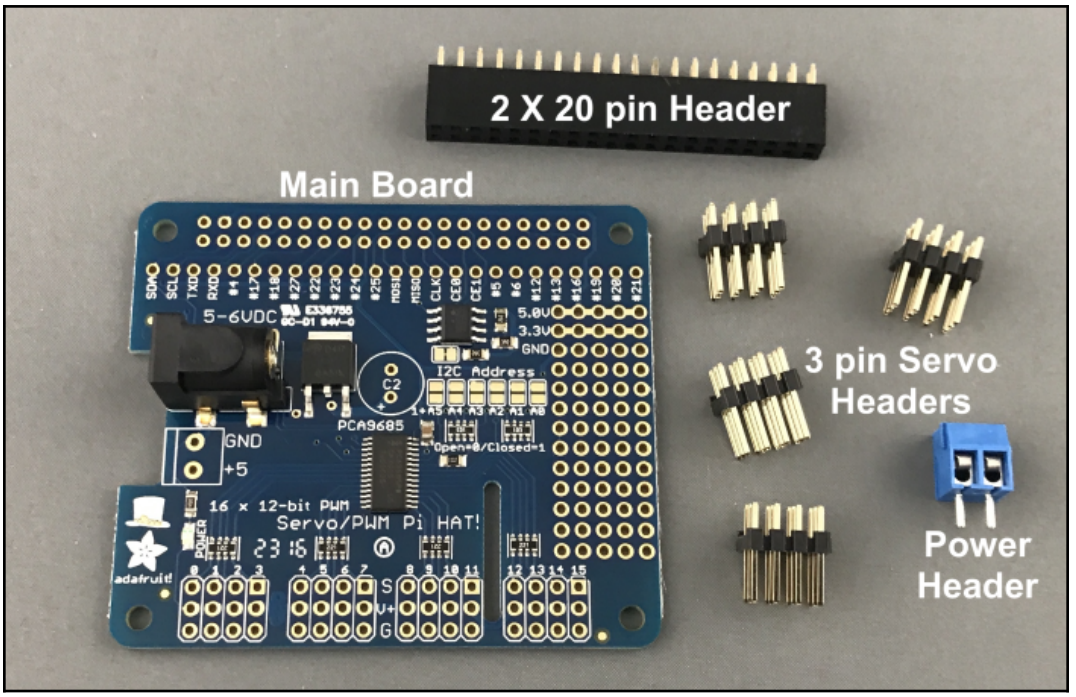


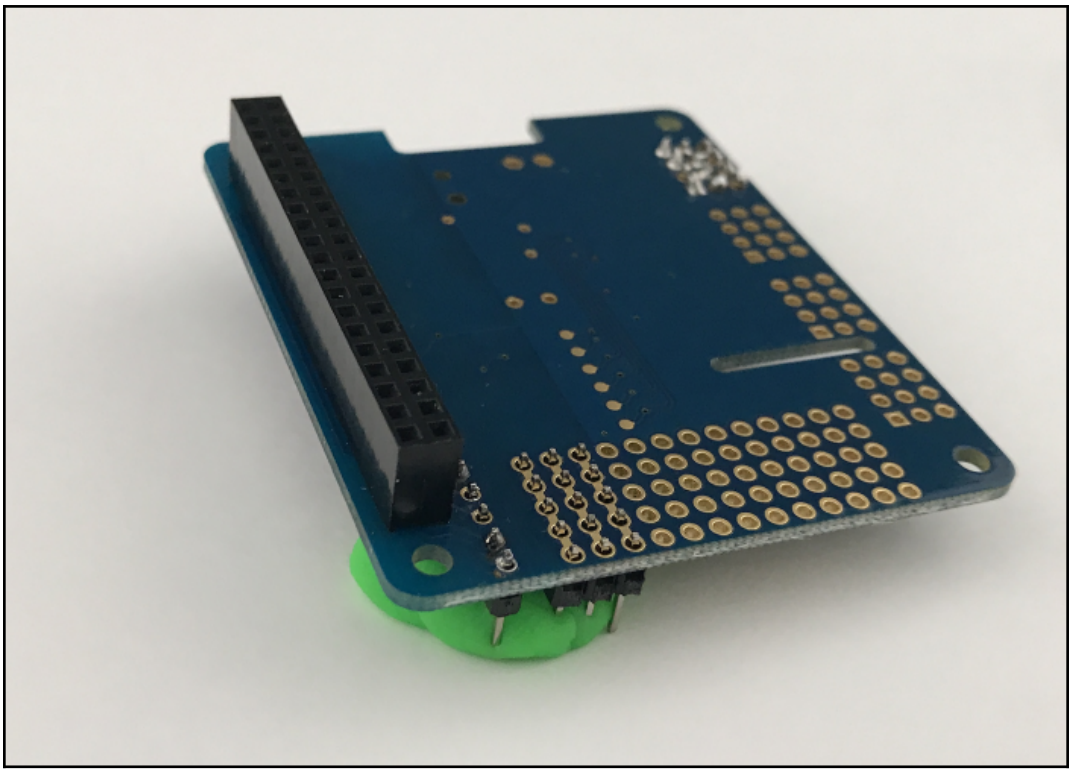

Rogers Wi-Fi
3:07 PM
100%
Text Message
Today 11:26 AM
Sent from your Twilio trial account - There is someone at the door
Text Message

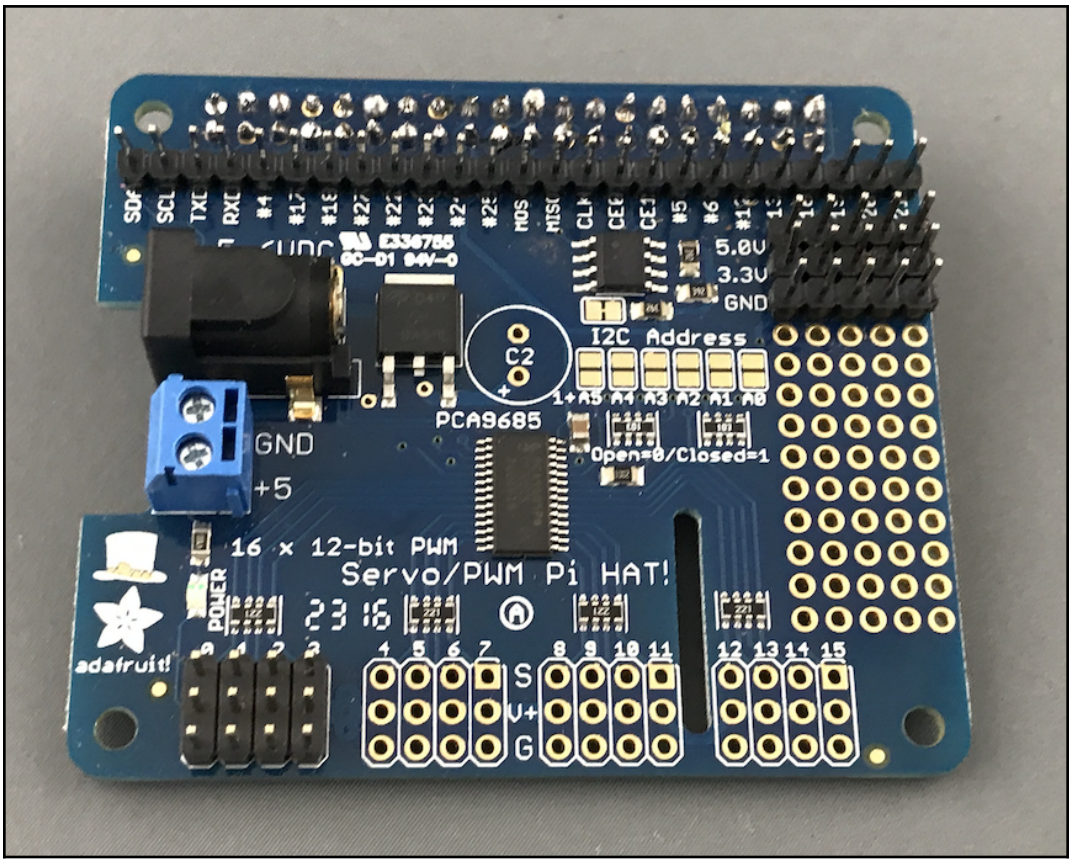

Rogers Wi-Fi
3:07 PM
100%
Today 2:40 PM
Sent from your Twilio trial account - Josephine is at the door!
Text Message

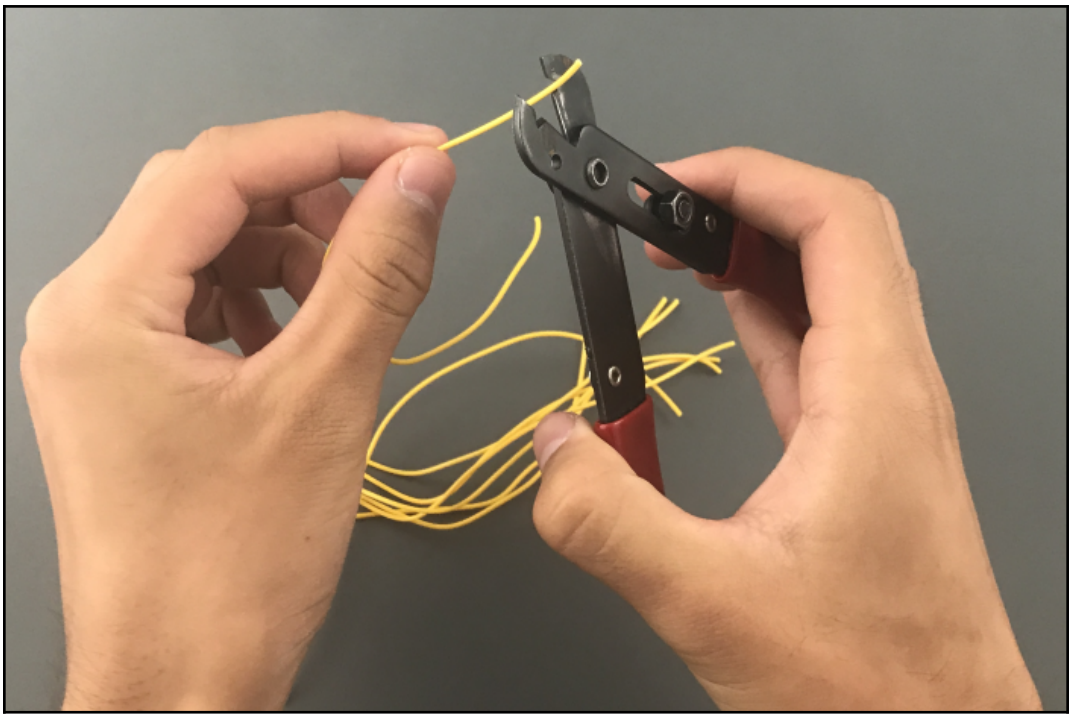
Chapter 22: Introducing the Raspberry Pi Robot Car

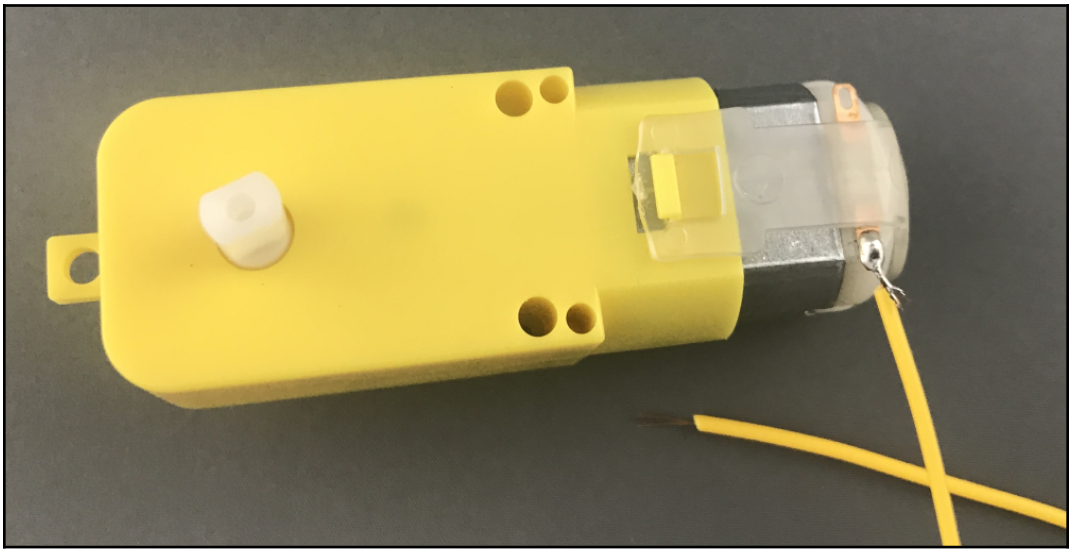


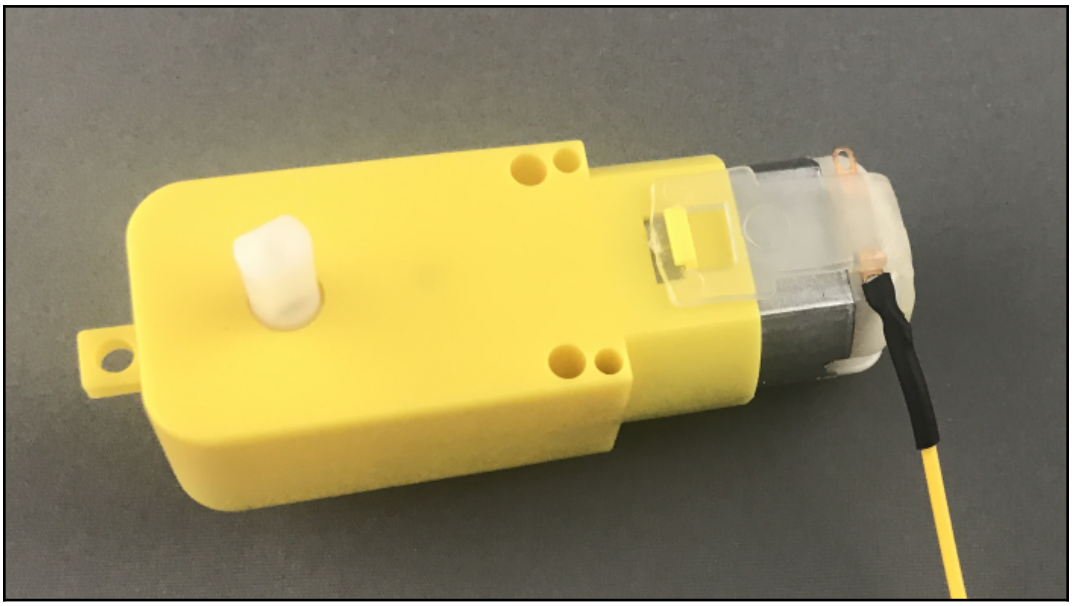

2 X 20 pin Header
Main Board
3 pin Servo Headers
Power Header
5-6VDC
GND
+5
PCA9685
16 x 12-bit PWM
Servo/PWM Pi HAT!
adafruit!
I2C Address
5.0V
3.3V
GND

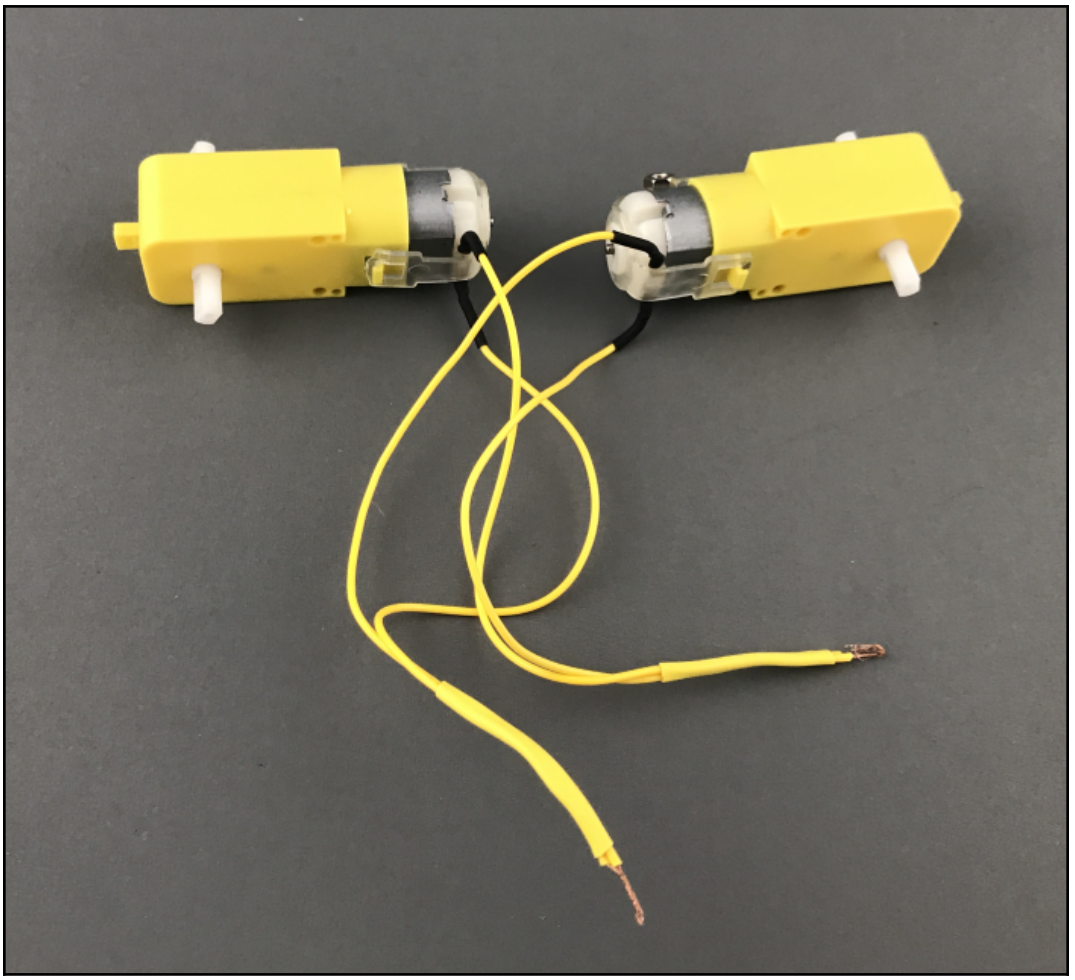


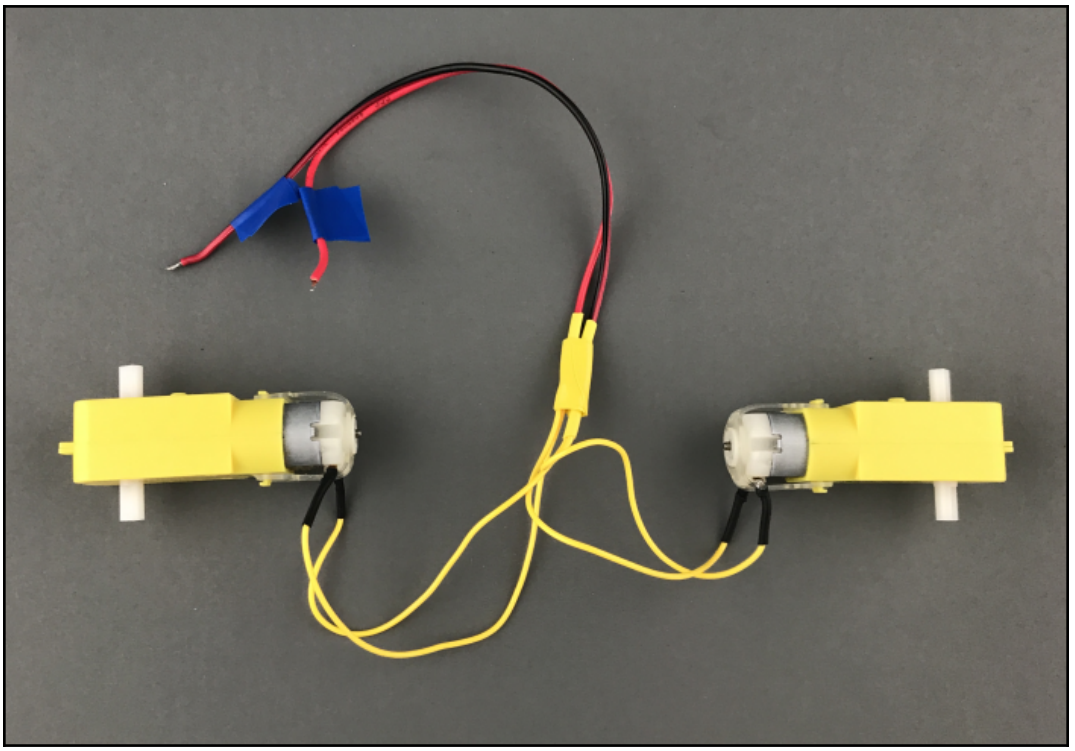

5.0V
3.3V
GND
I2C Address
A5 A4 A3 A2 A1 A0
Open=0/Closed=1
C2
PCA9685
GND
+5
16 x 12-bit PWM
Servo/PWM Pi HAT!
POWER
adafruit!
S
V+
G

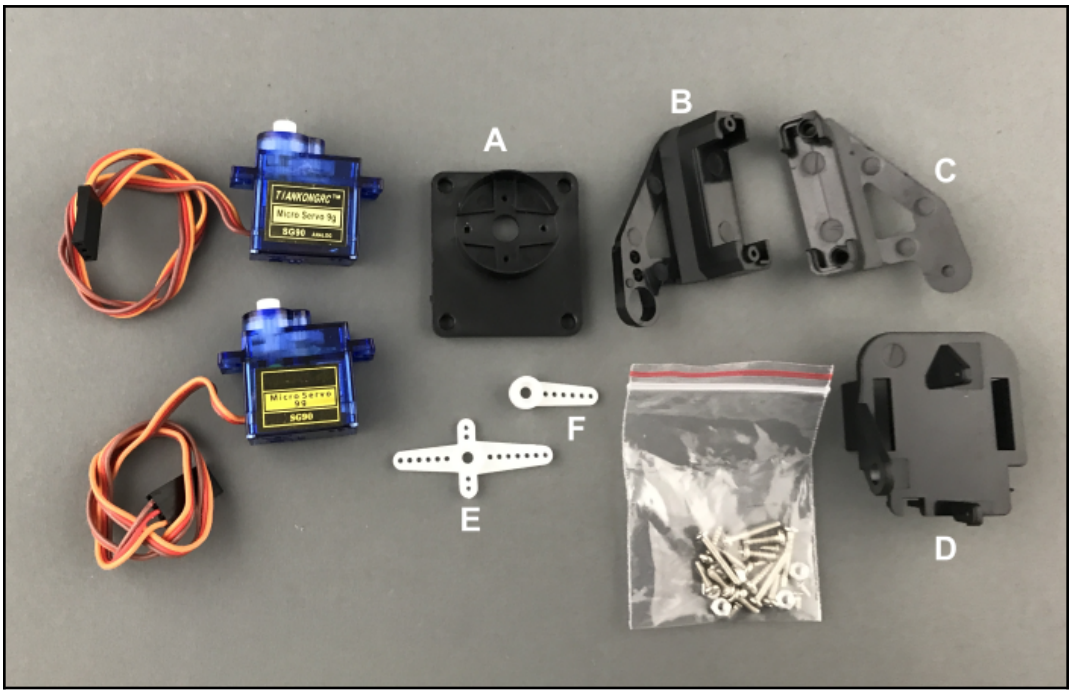


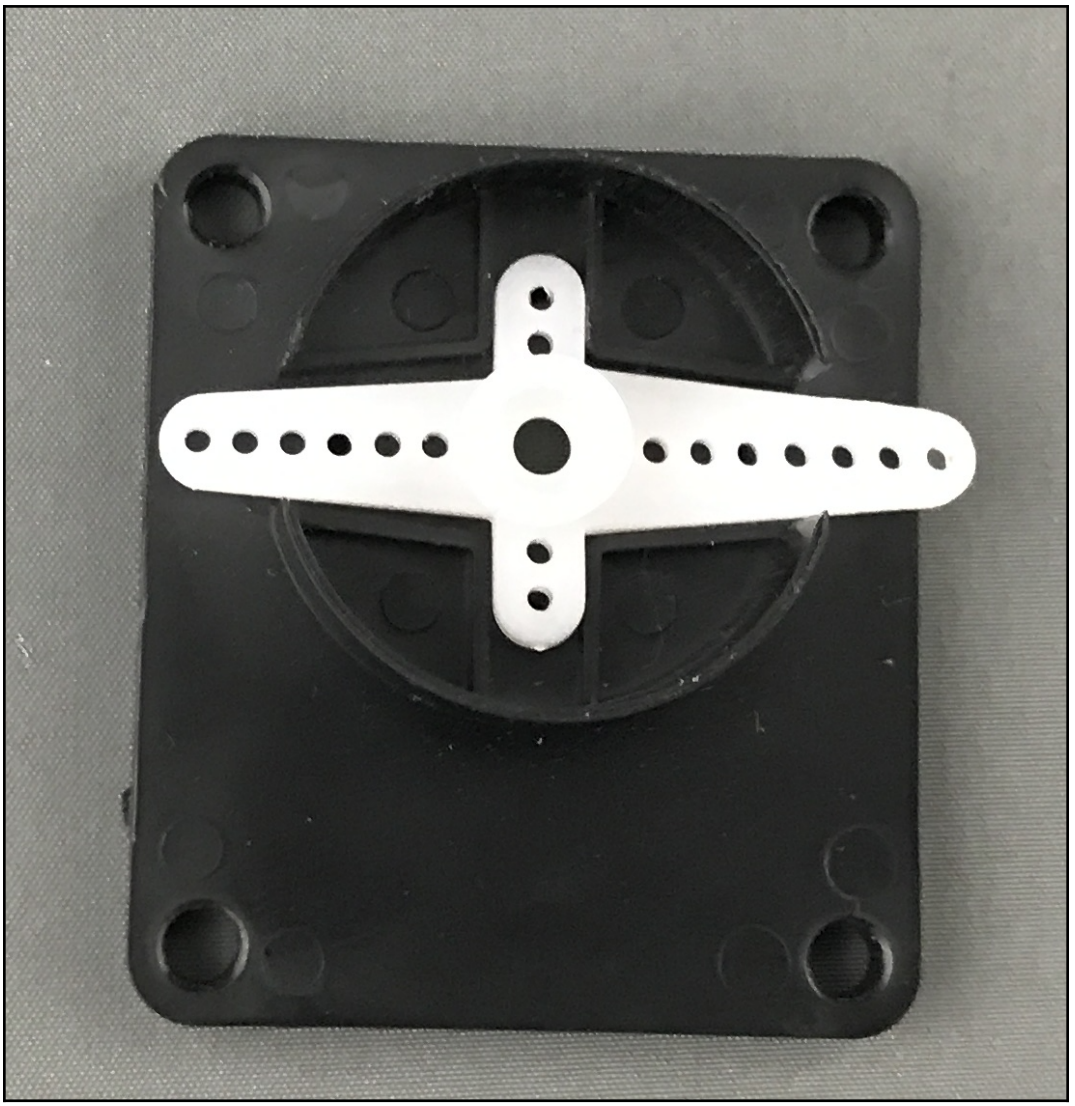


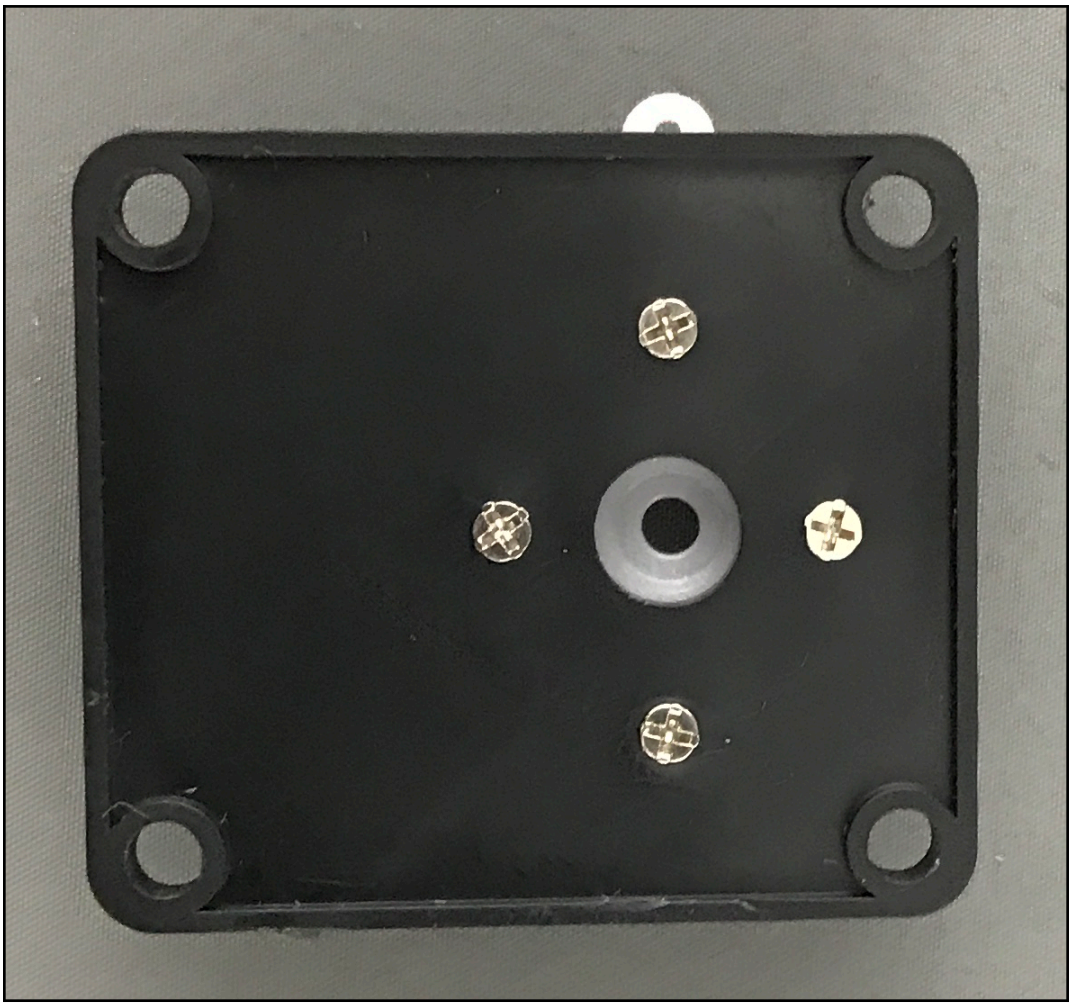


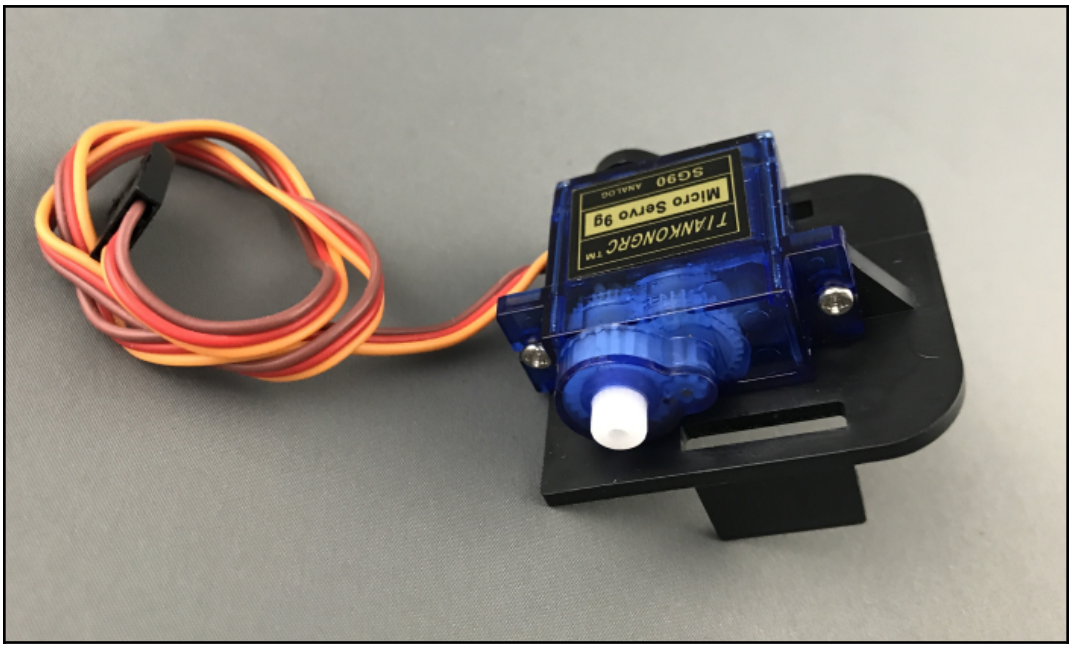


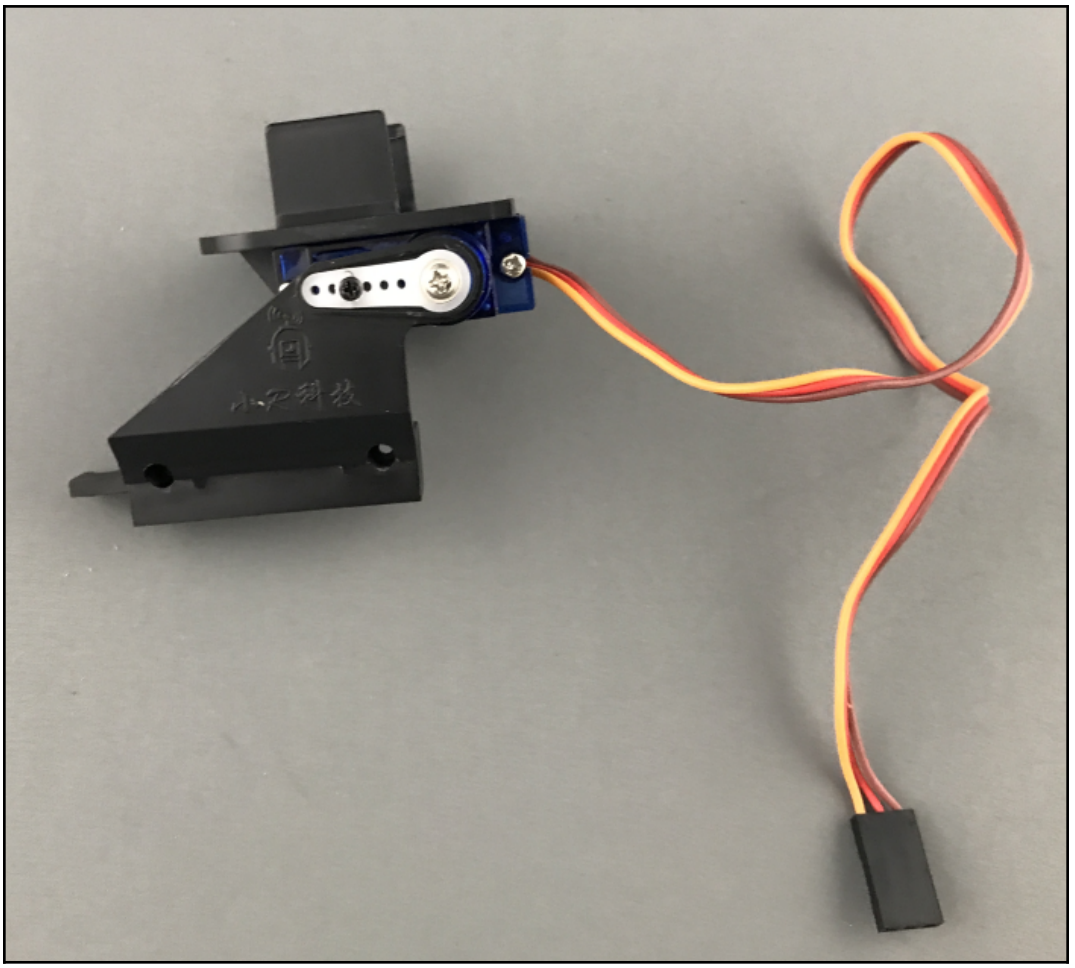


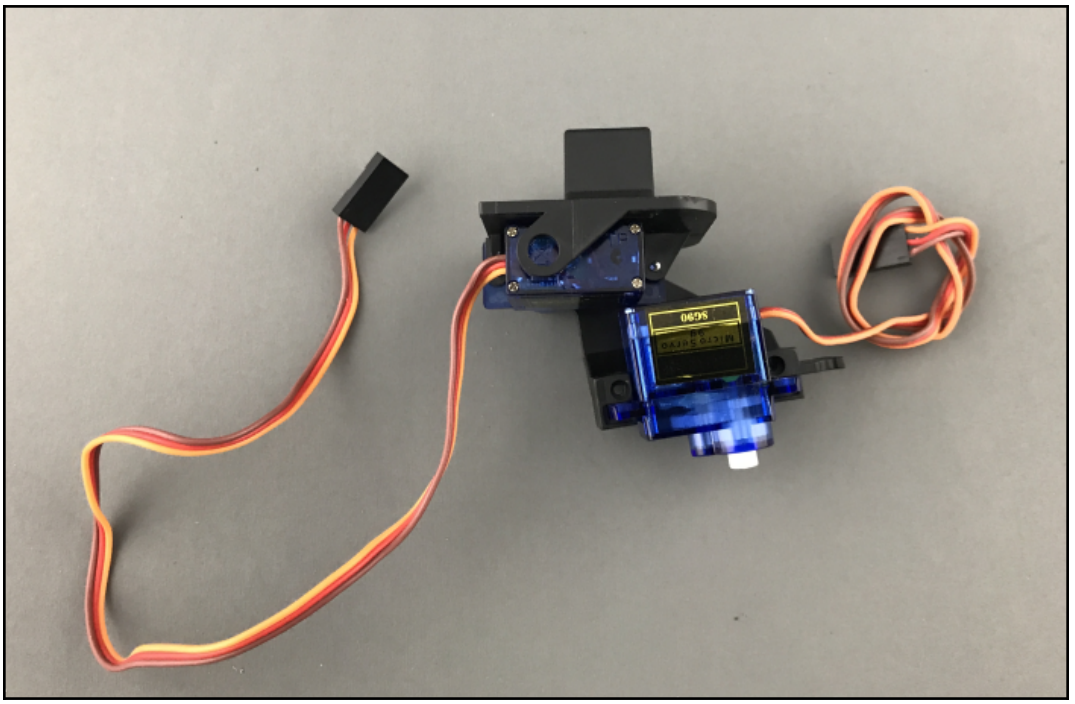

A
B
C
D
E
F
TIANKONGRC
Micro Servo 9g
SG90

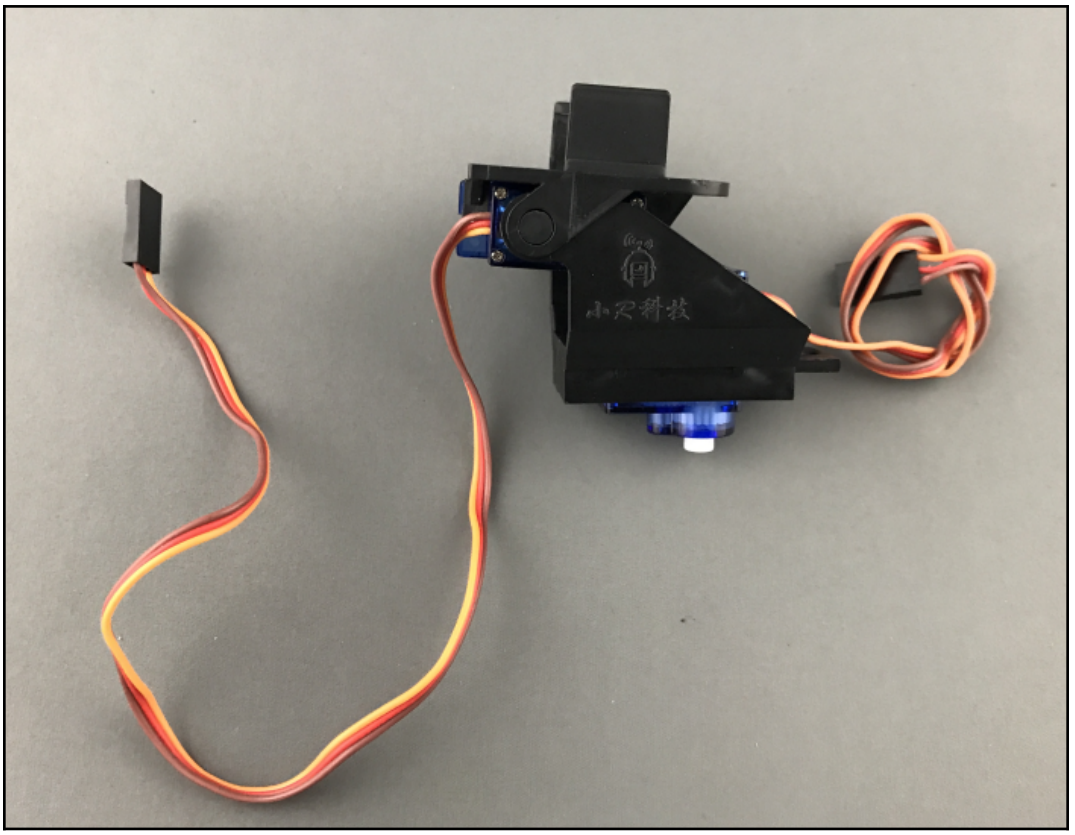


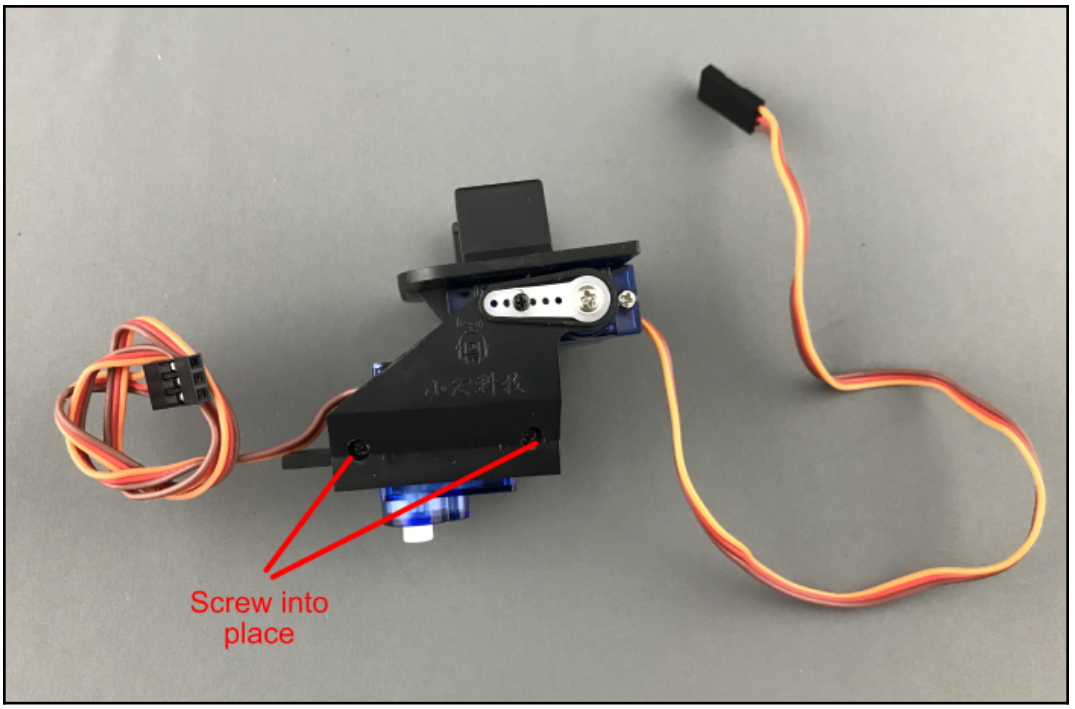


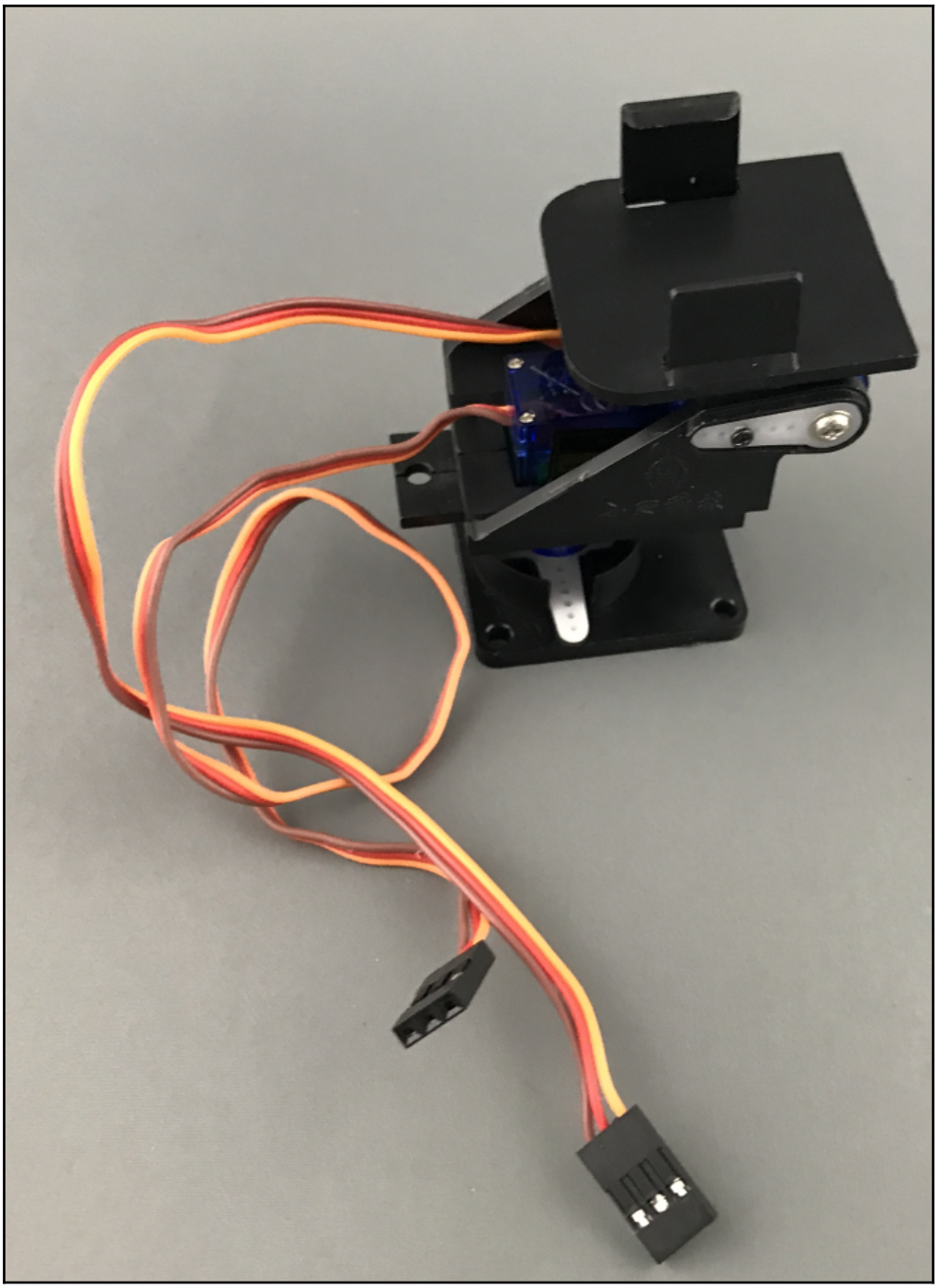

TIANKONGRC TM
Micro Servo 9g
SG90 ANALOG

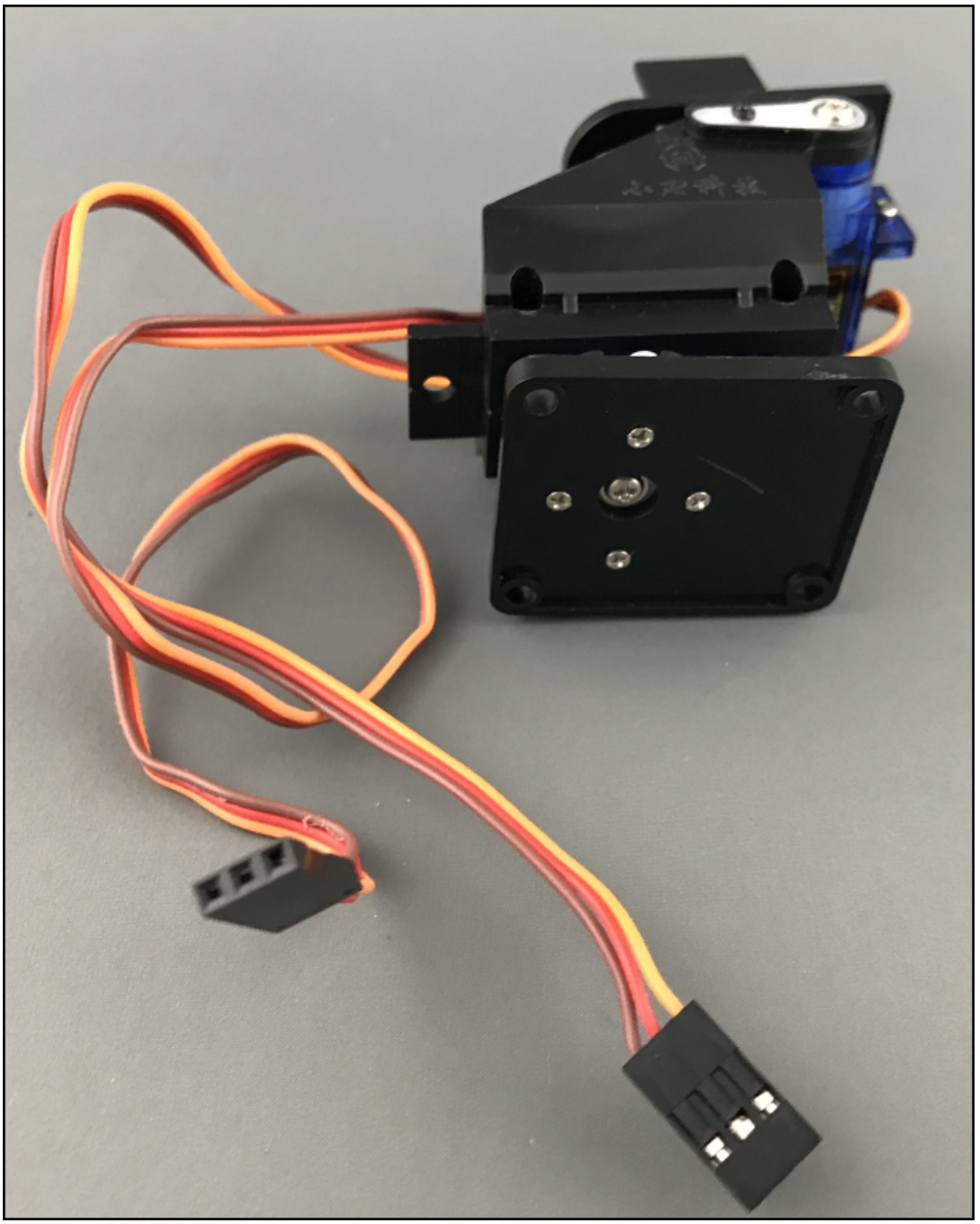


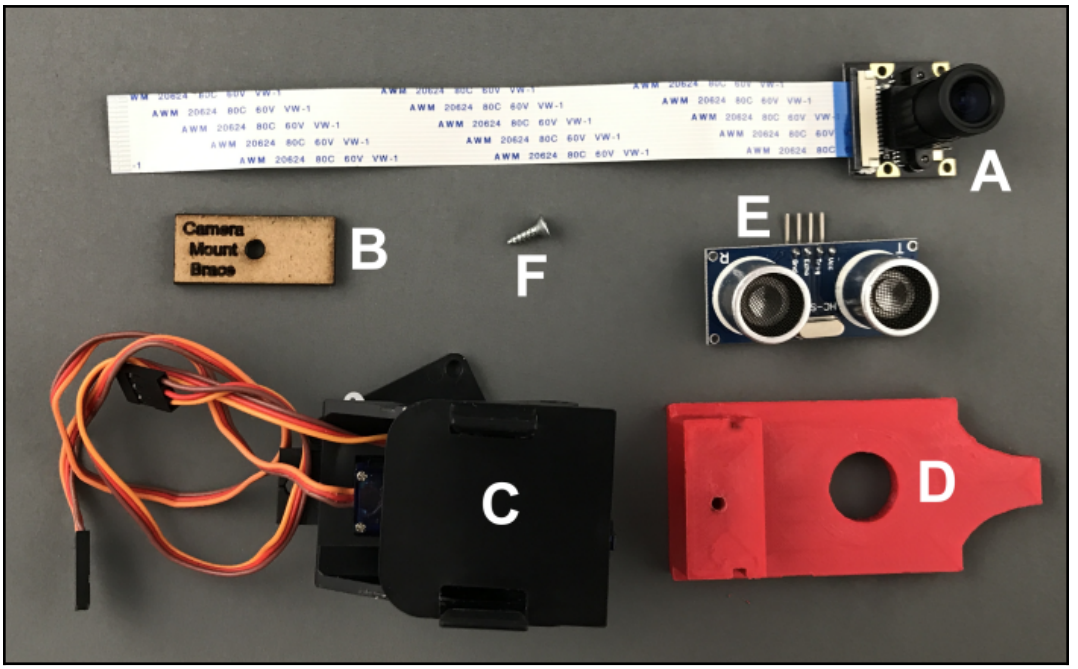

9690

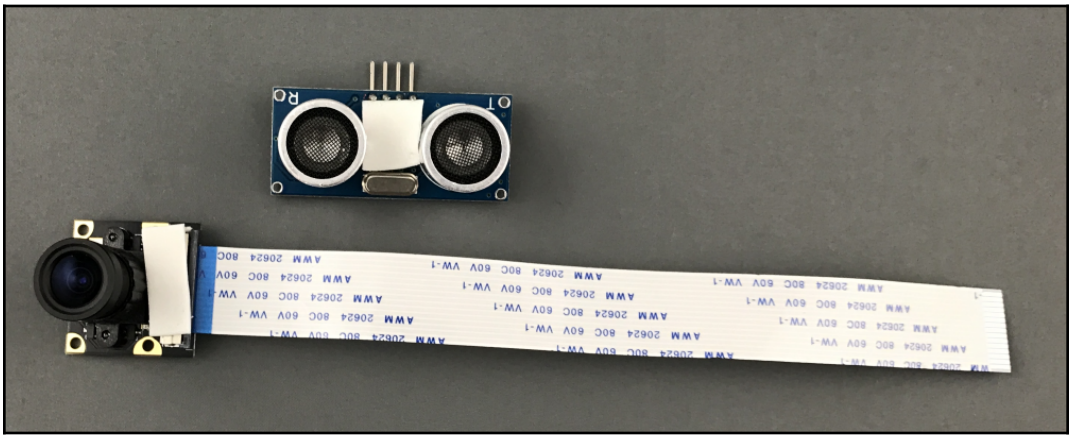


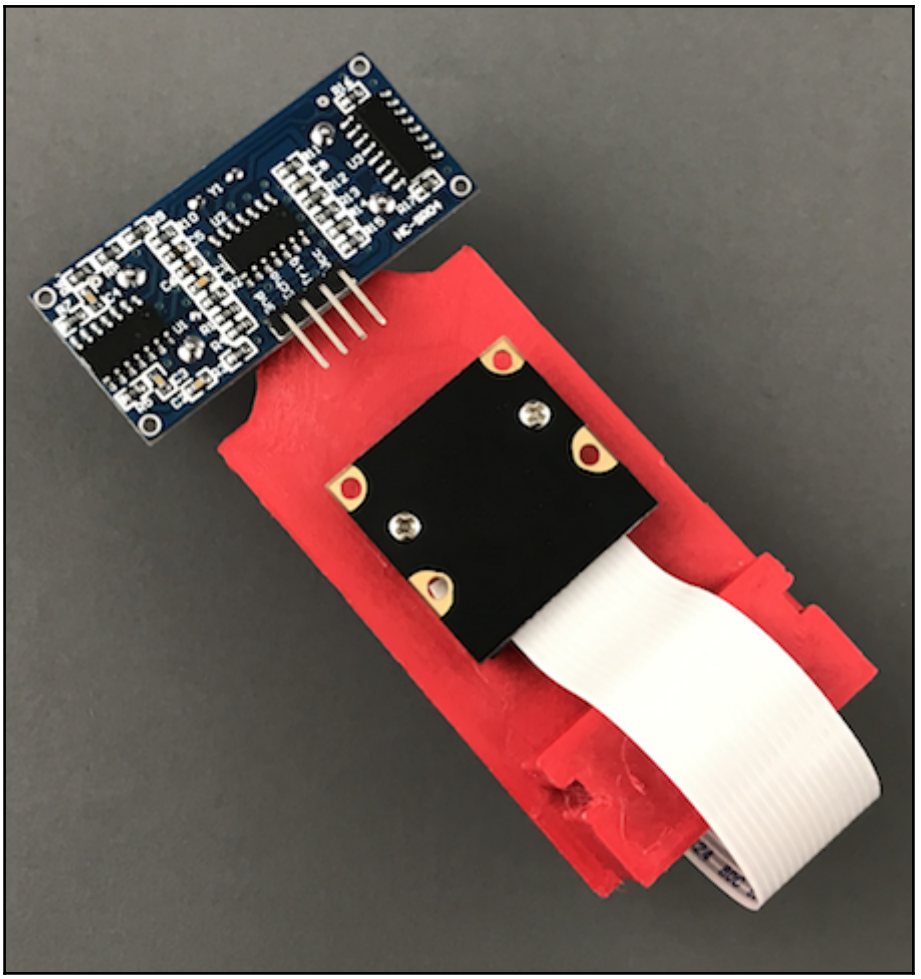

Screw into
place

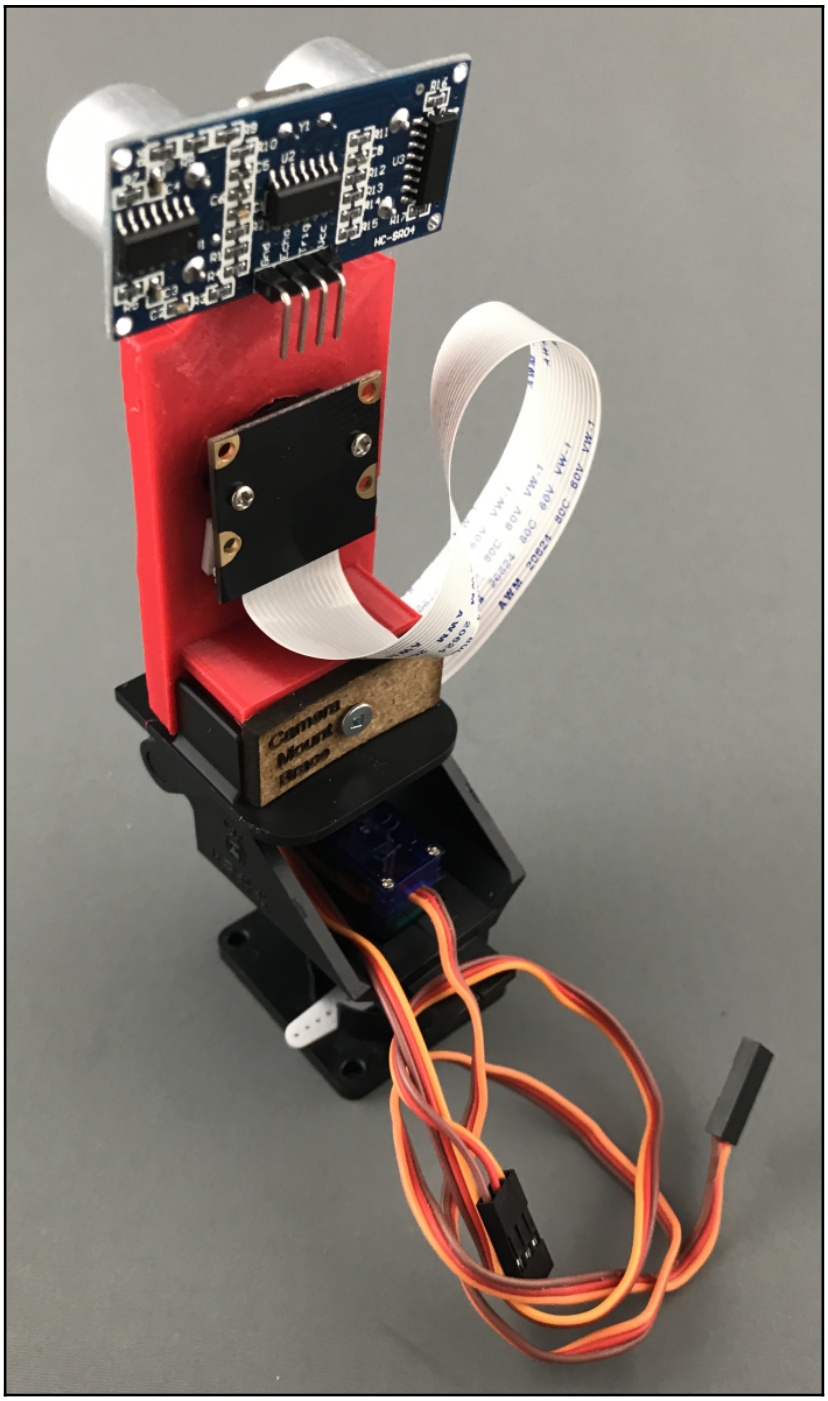


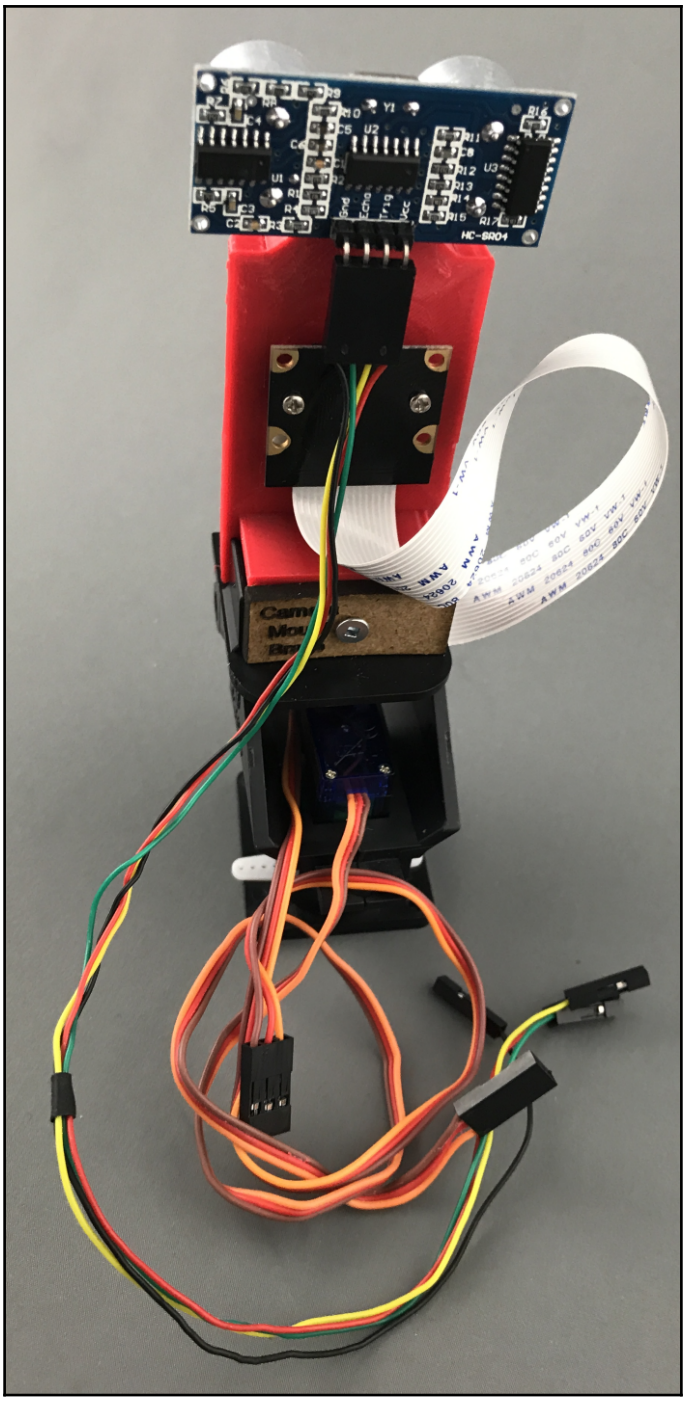


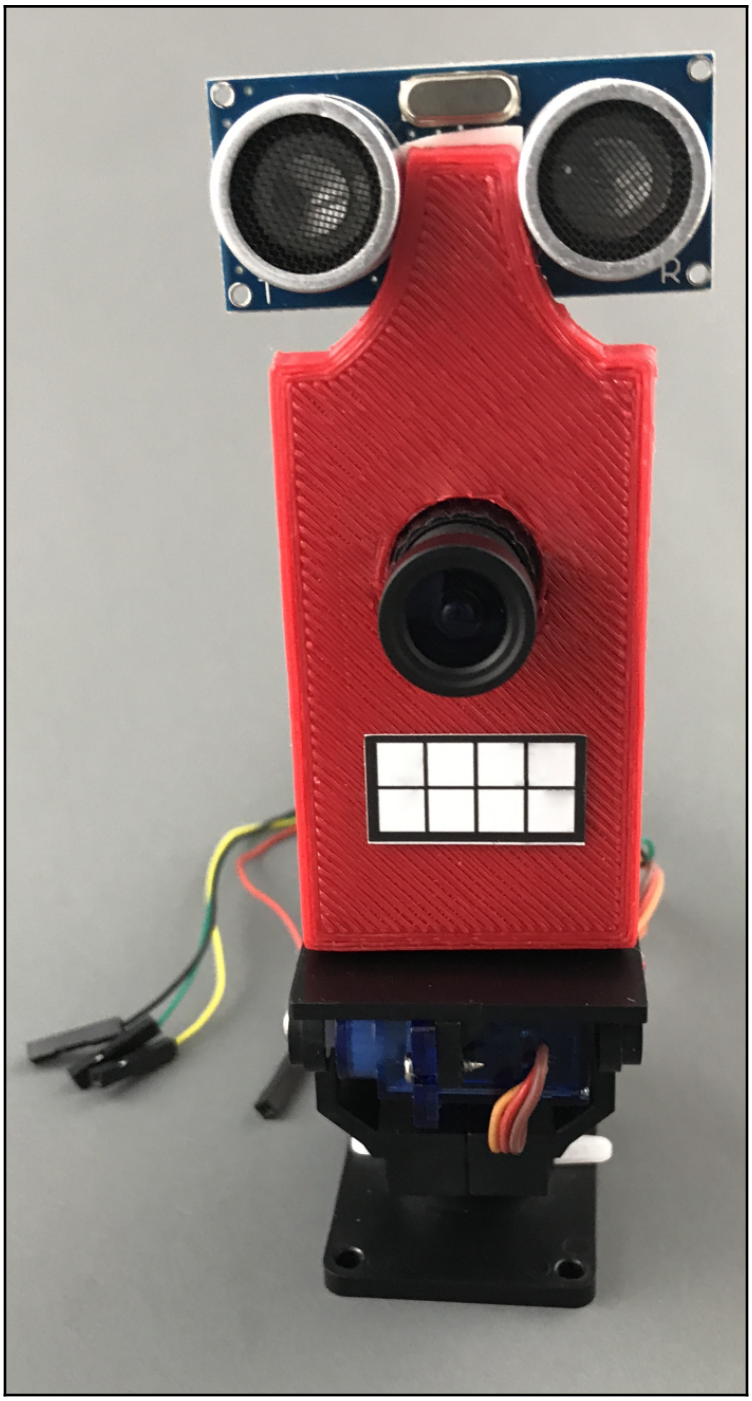

A
Camera
Mount
Brace
B
F
E
C
D

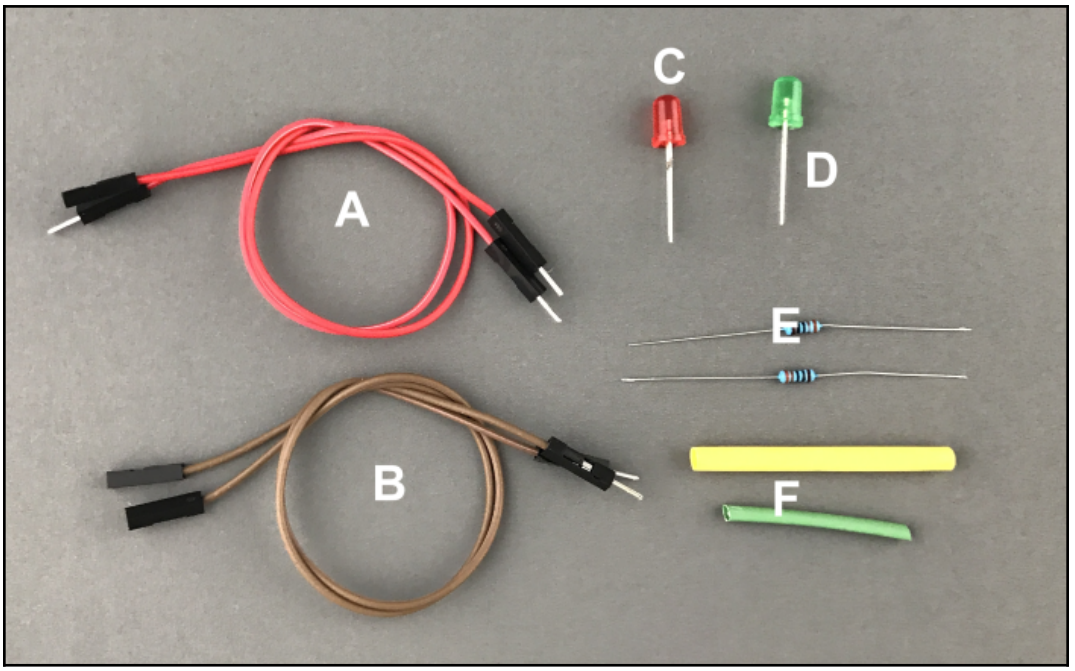


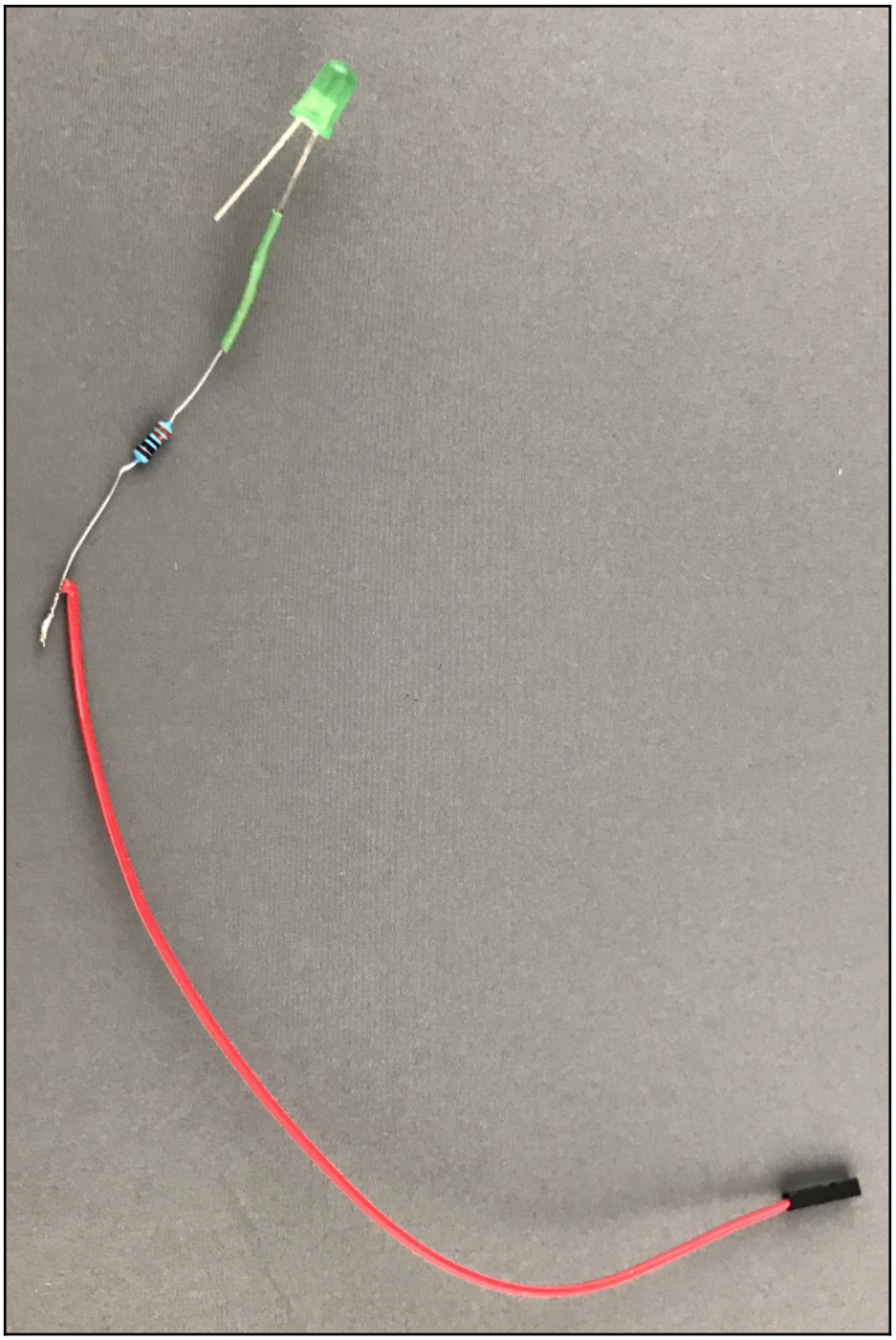


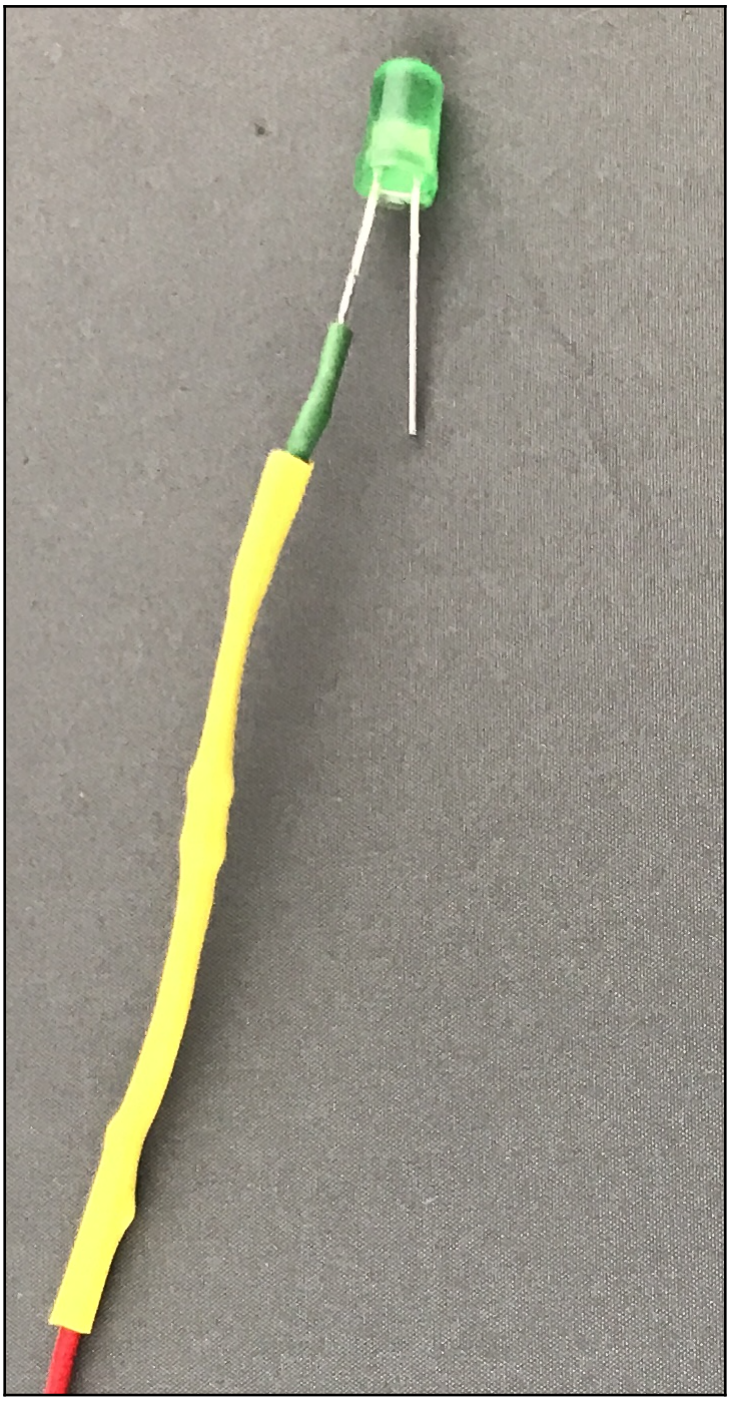

Camera
Mount
Brace

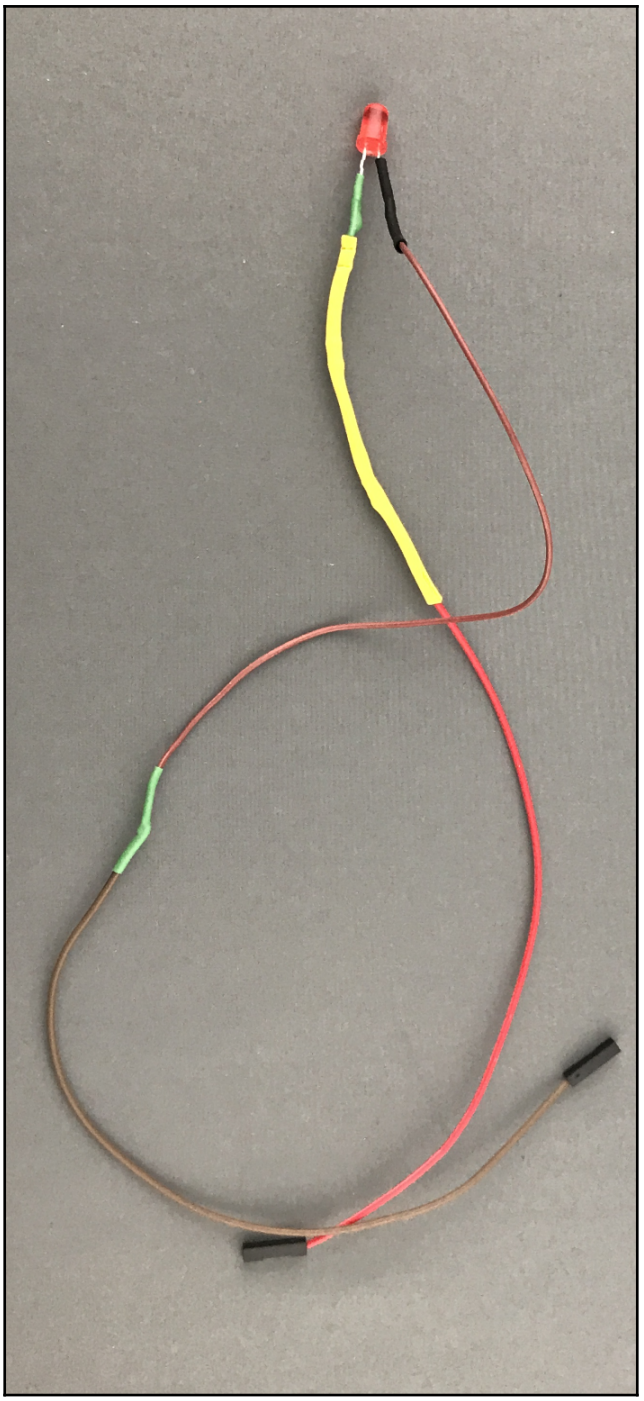

HC-SR04

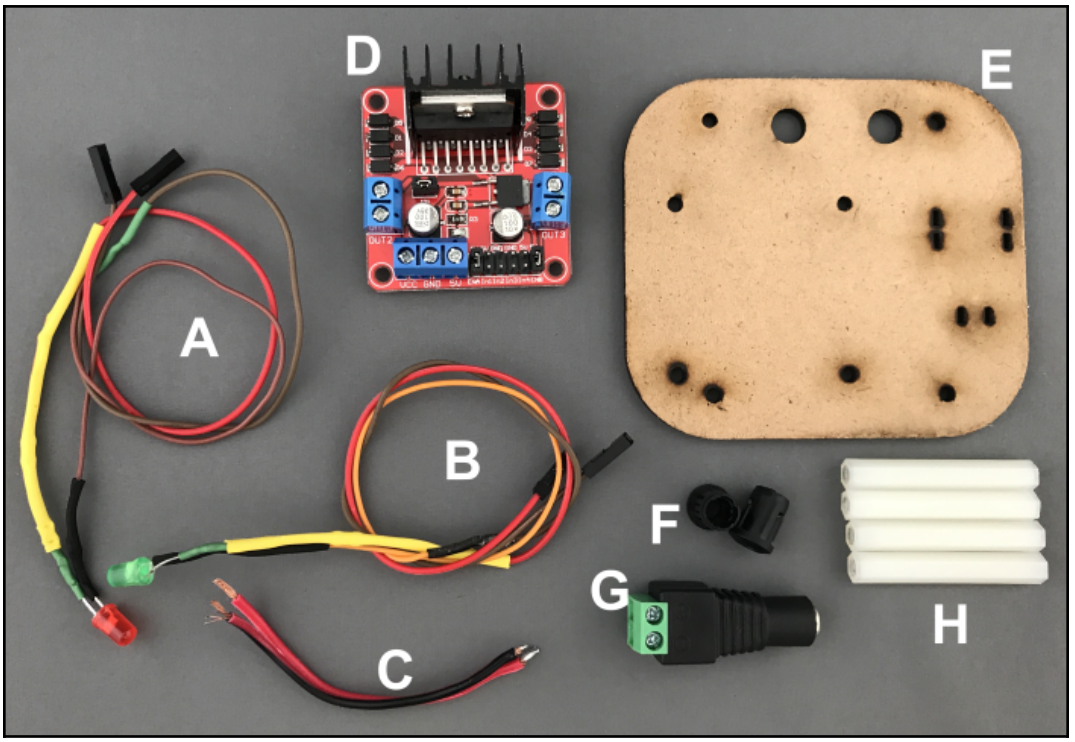

T
R

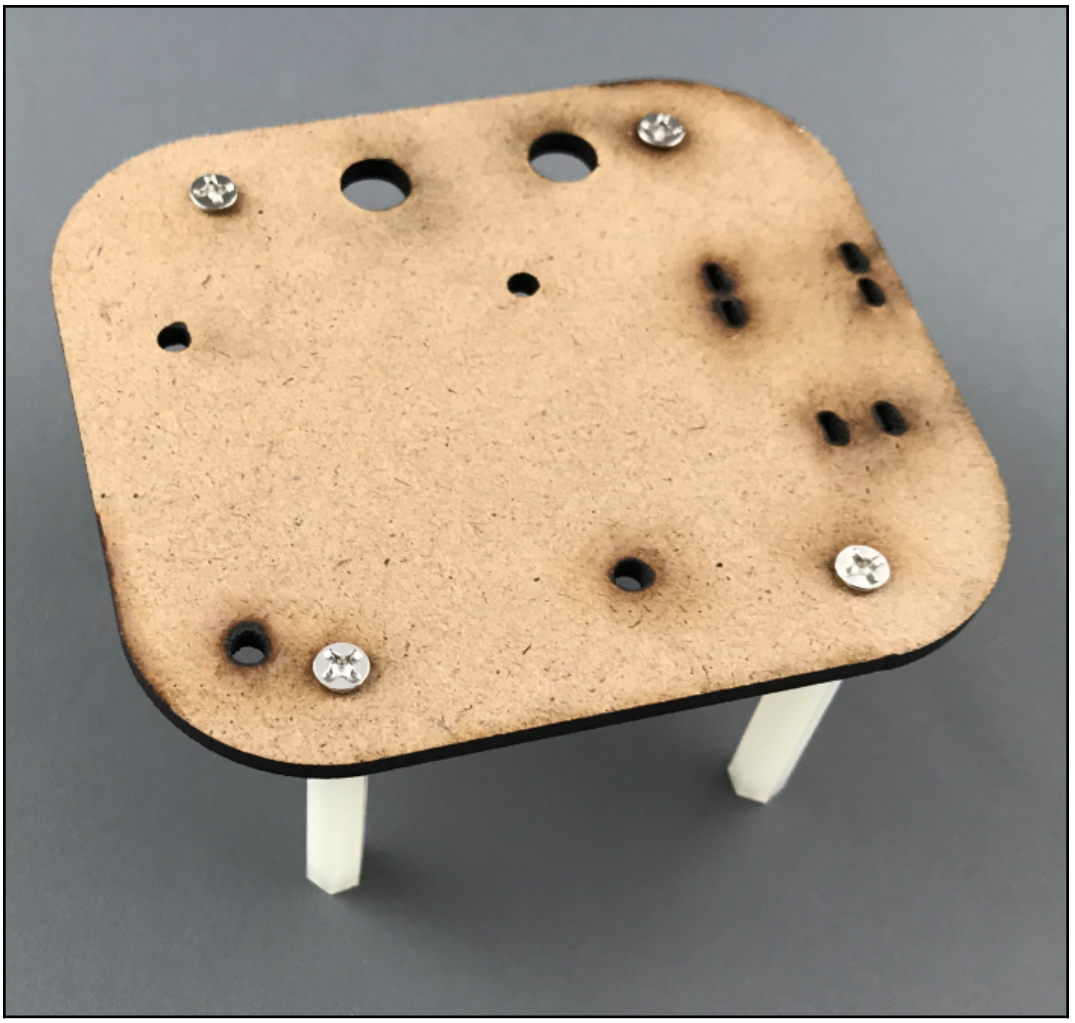

C
D
A
E
B
F

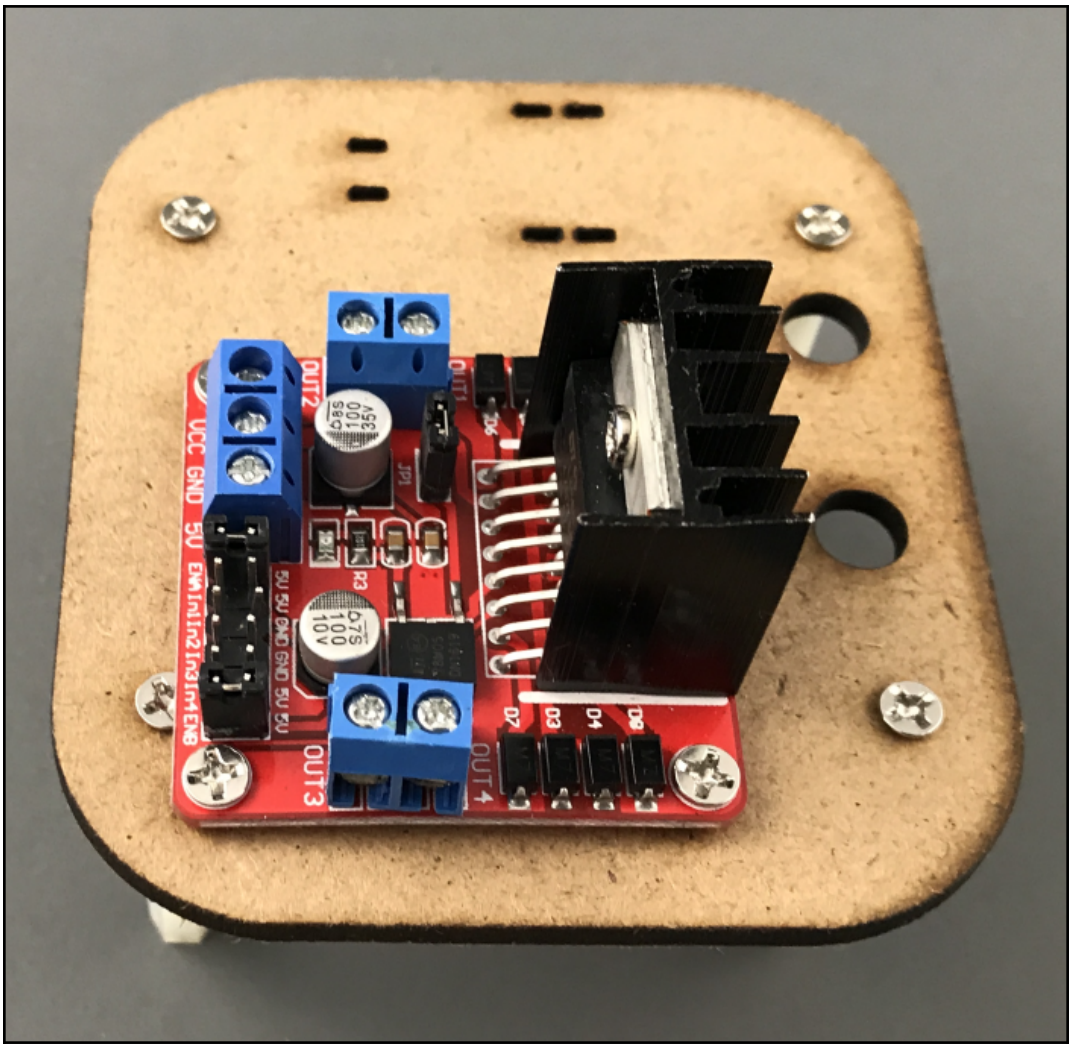


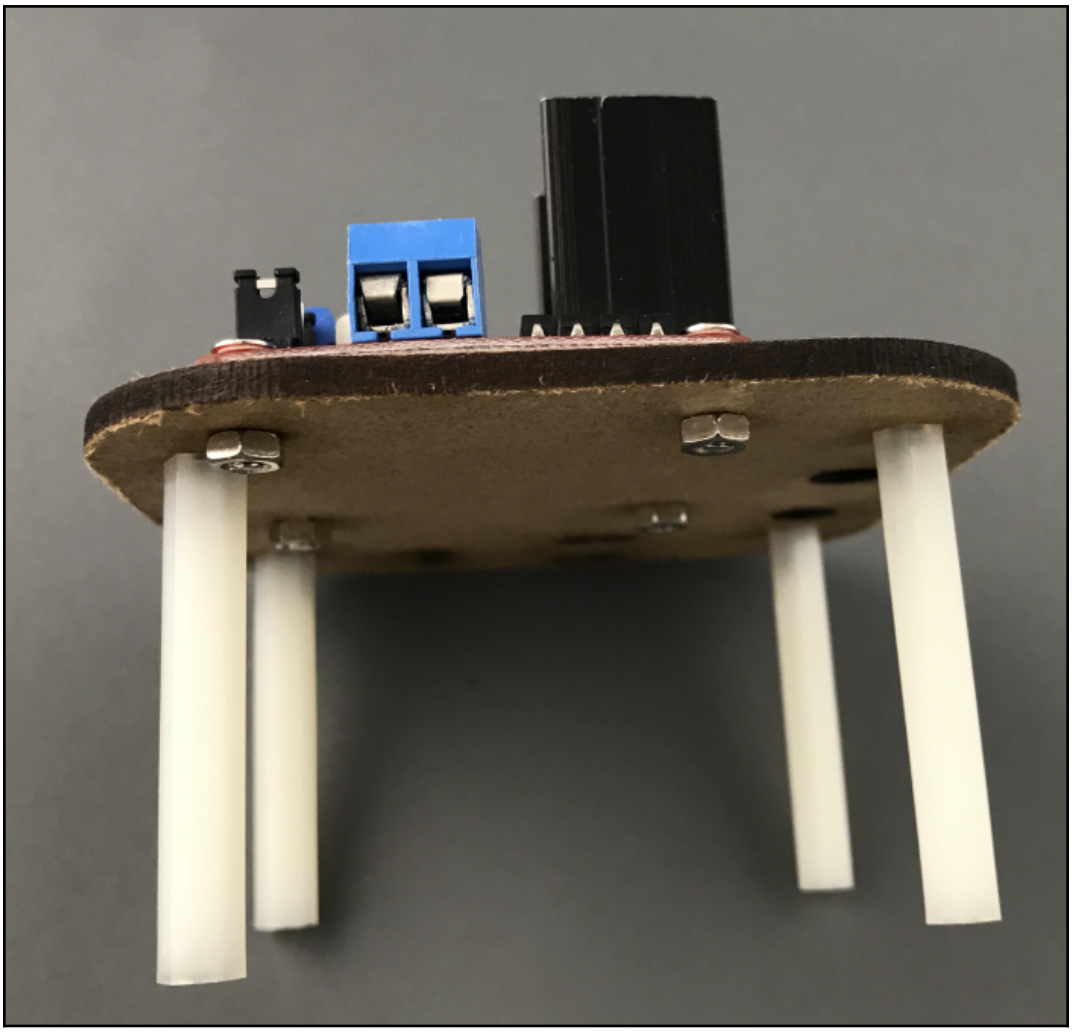


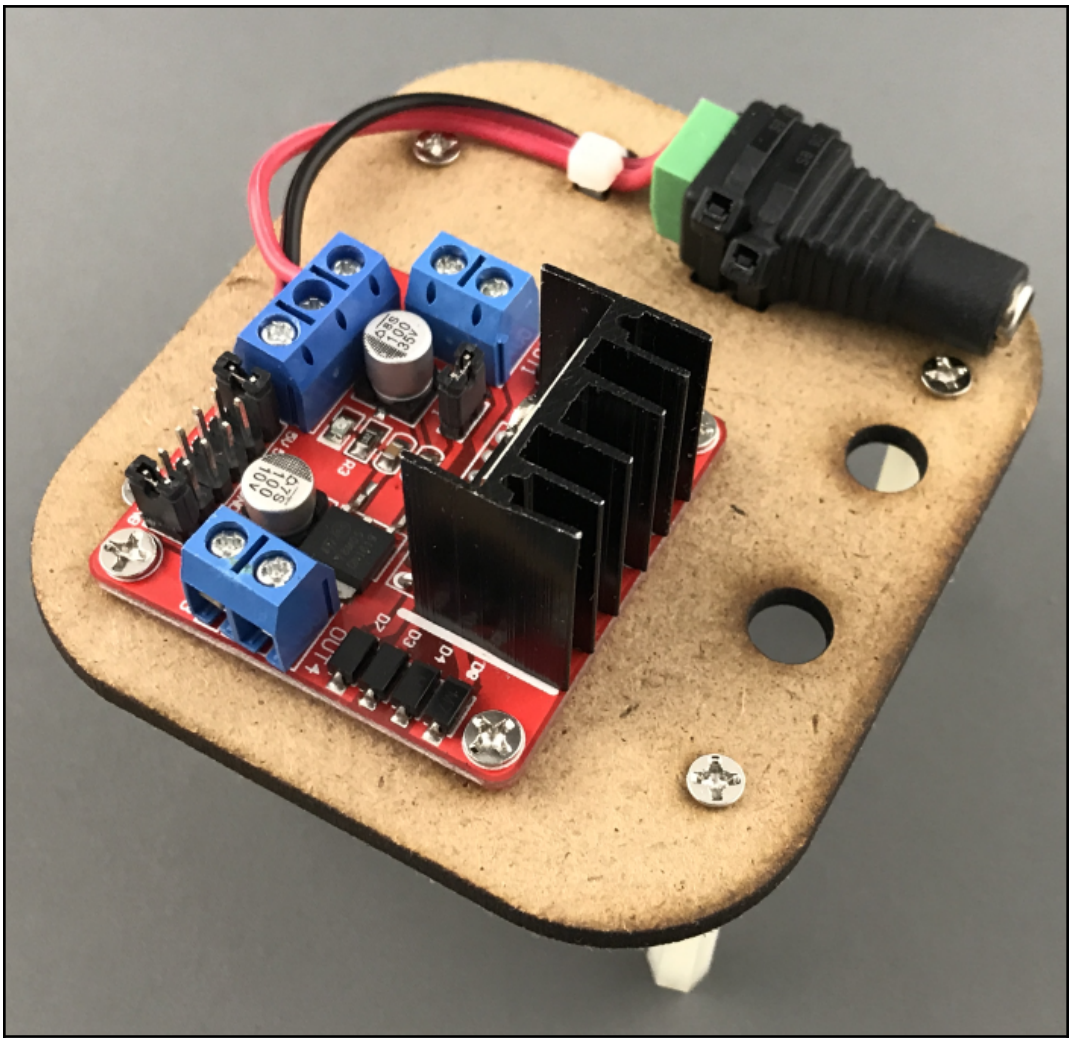


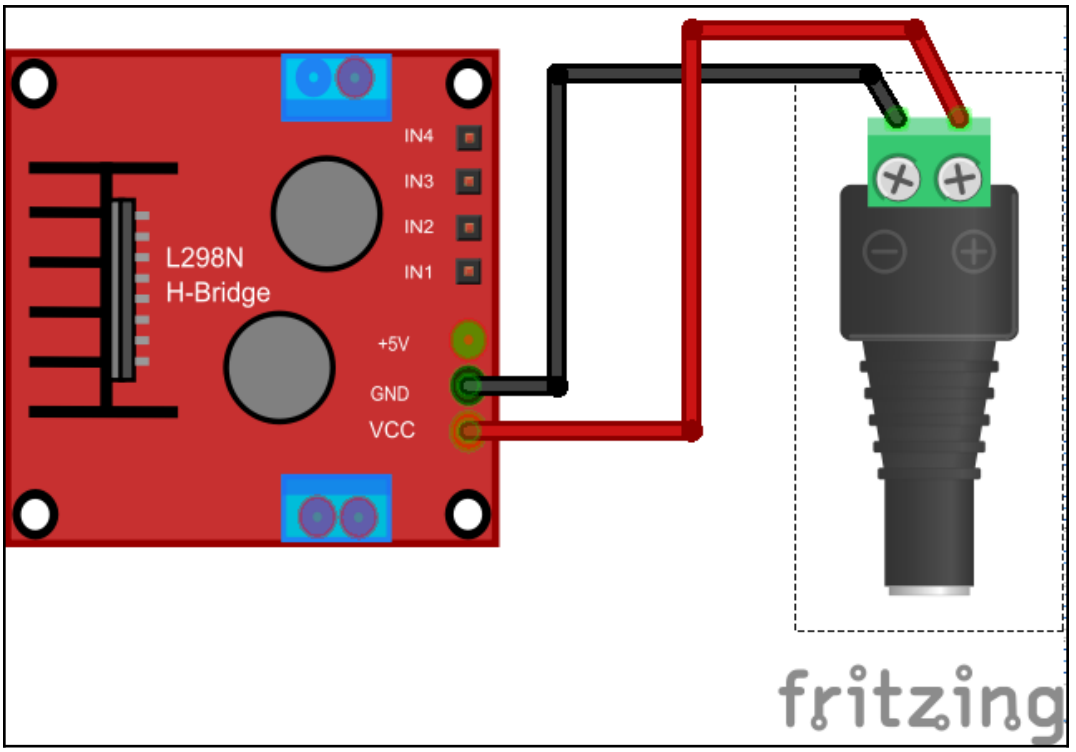

D
E
A
B
F
G
H
C

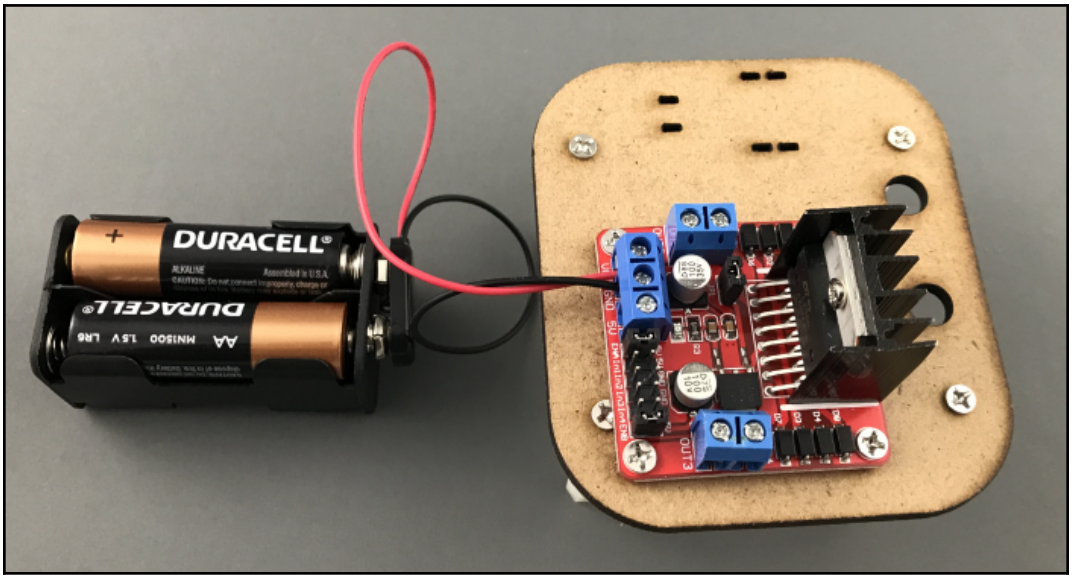


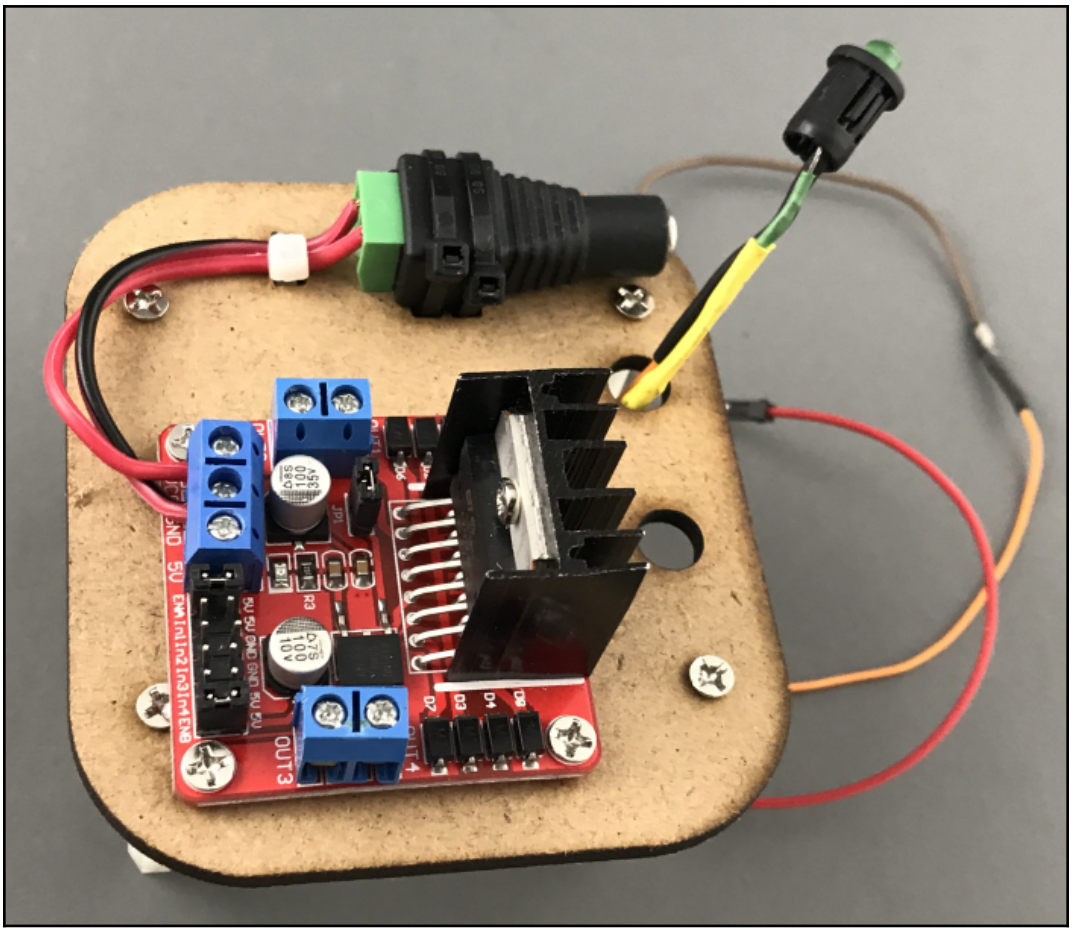

OUT2
OUT1
JP1
R3
OUT3
OUT4
VCC GND 5V ENA in1 in2 in3 in4 ENB
5V 5V GND GND 5V 5V

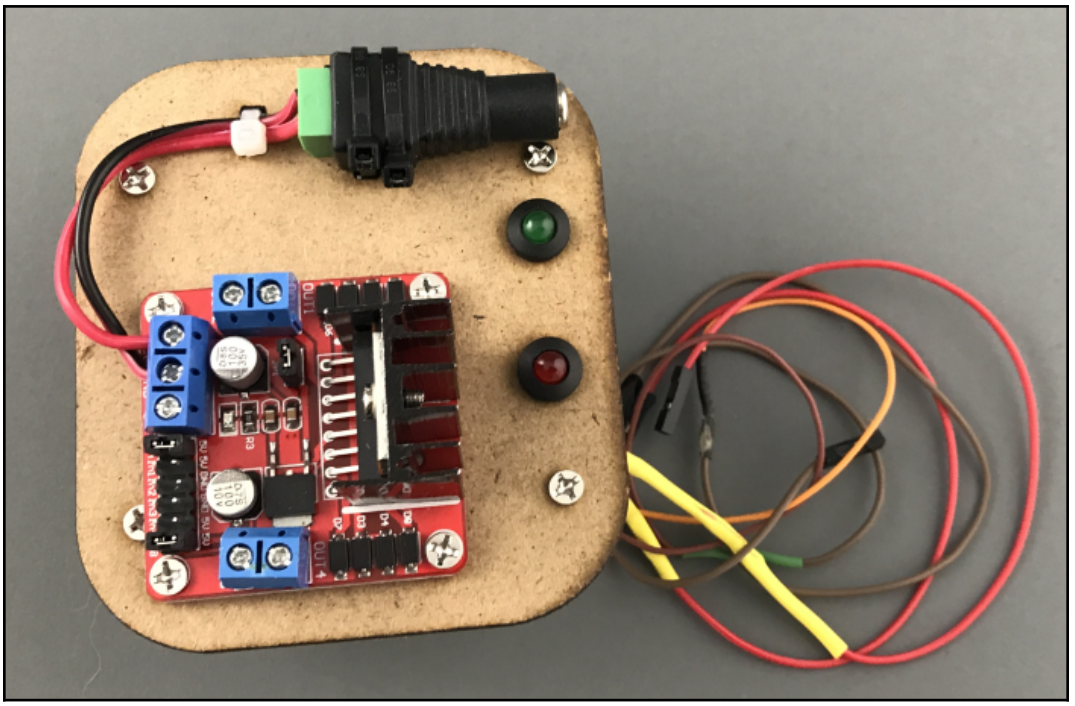


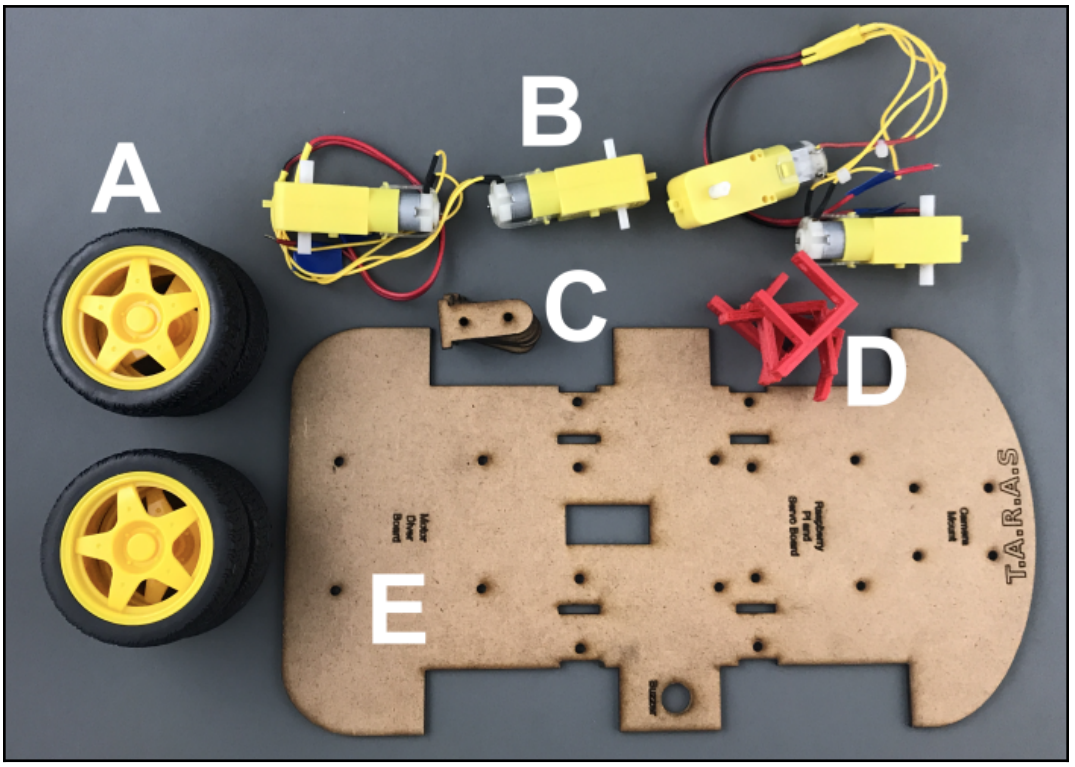


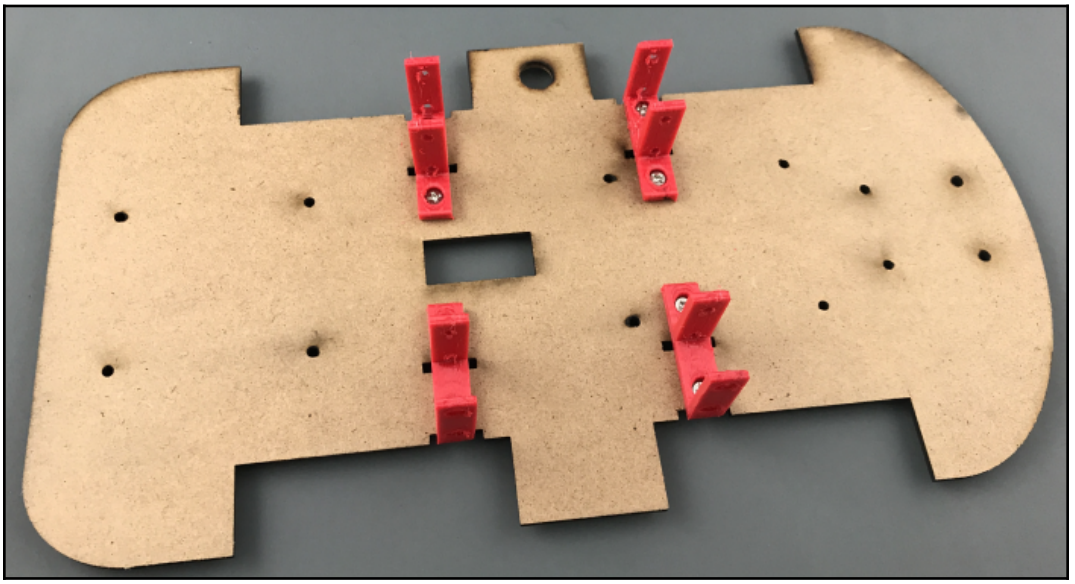

IN4
IN3
IN2
IN1
L298N
H-Bridge
+5V
GND
VCC
fritzing

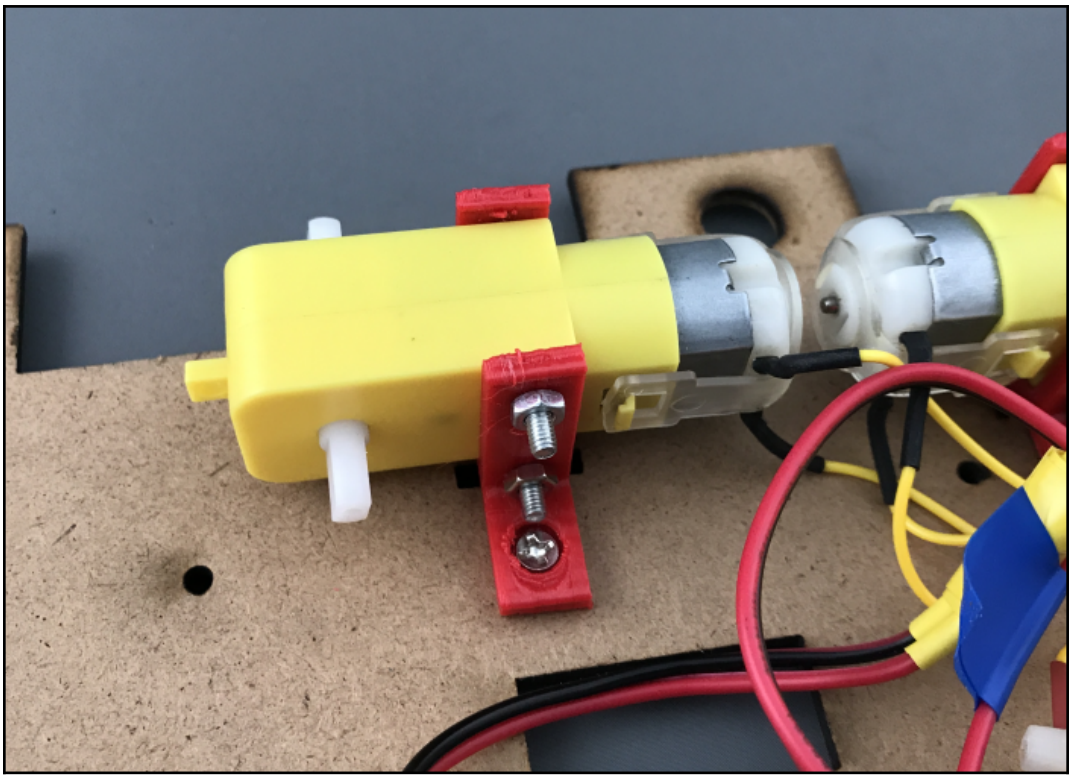

DURACELL

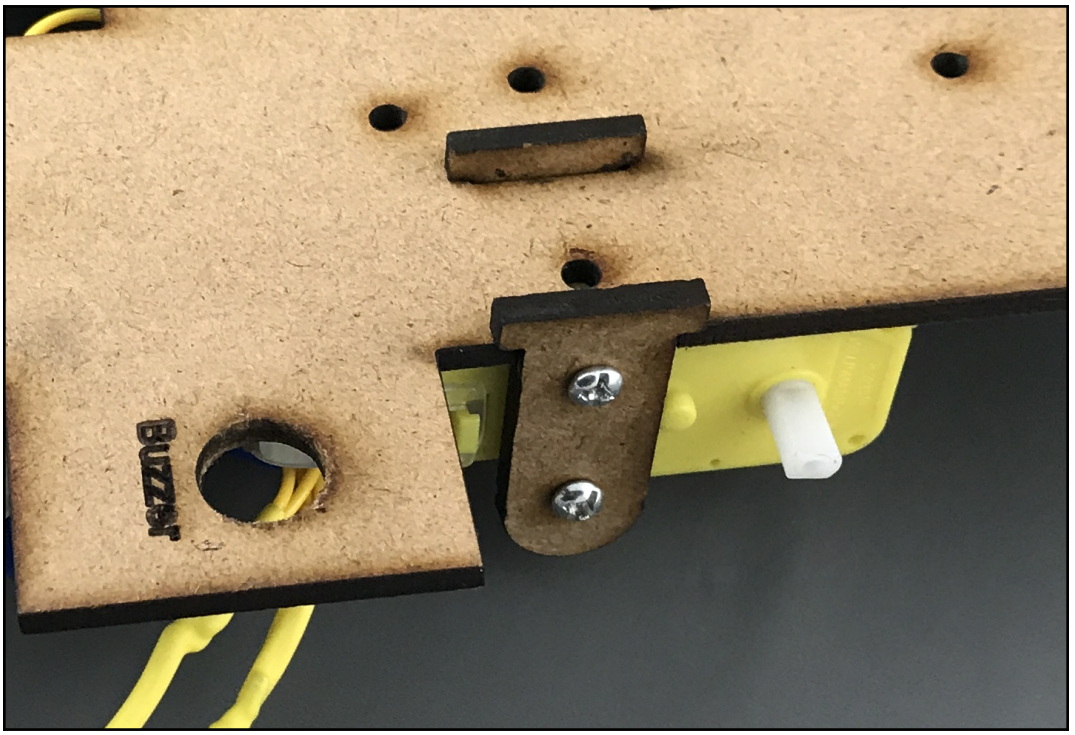

GND 5V ENA IN1 IN2 IN3 IN4 ENB
OUT3

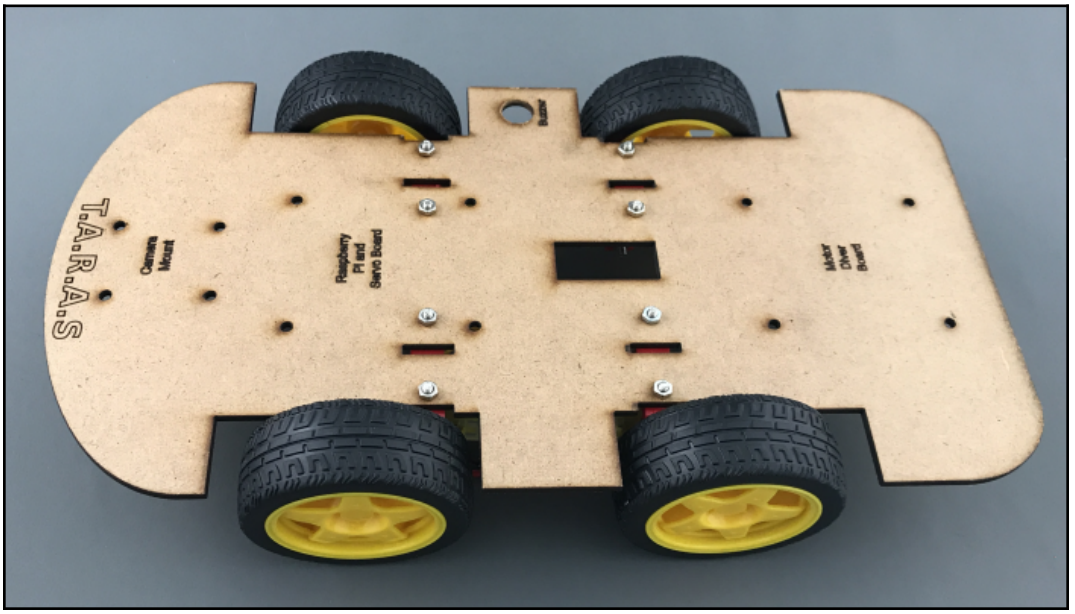


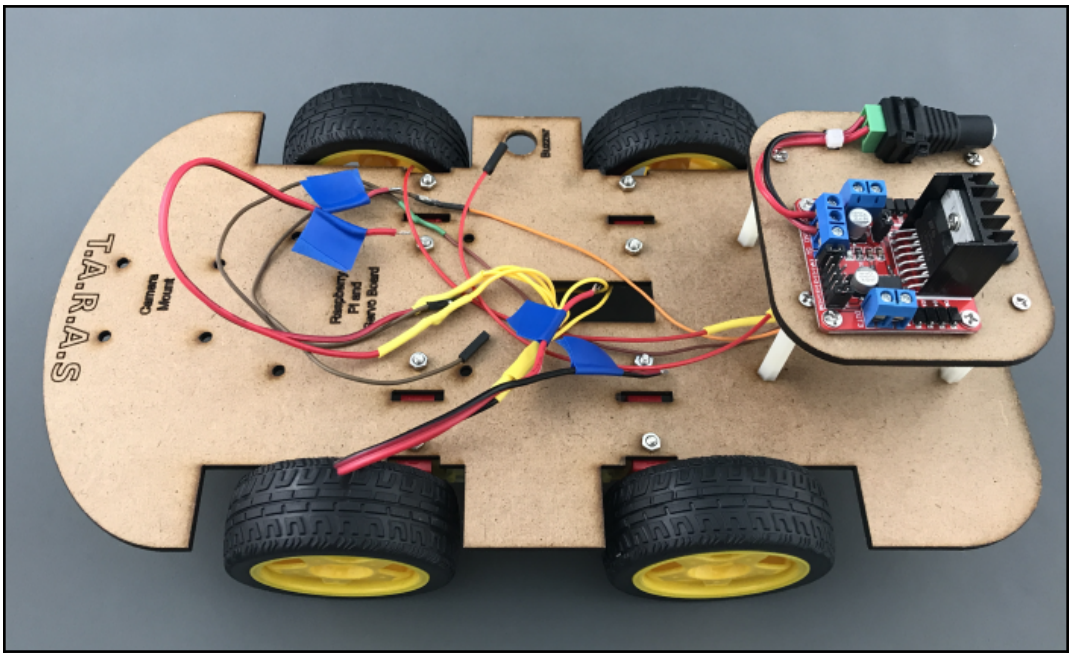

A
B
C
D
E
T.A.R.A.S

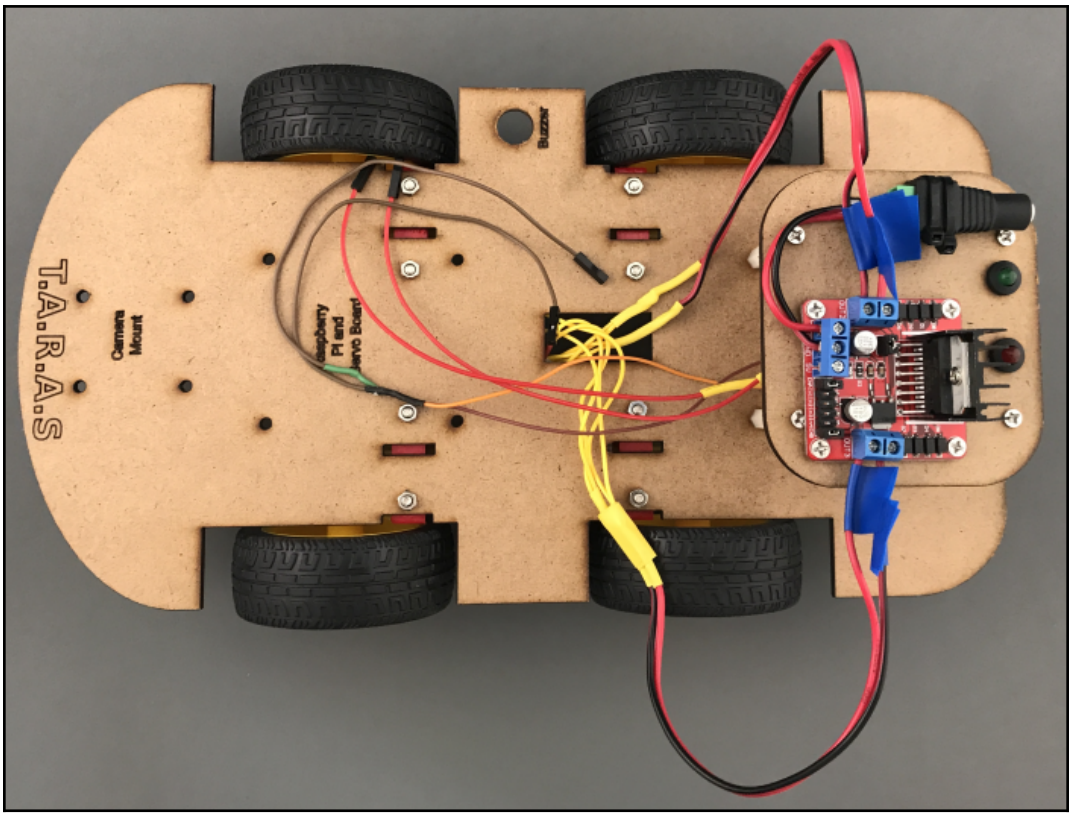


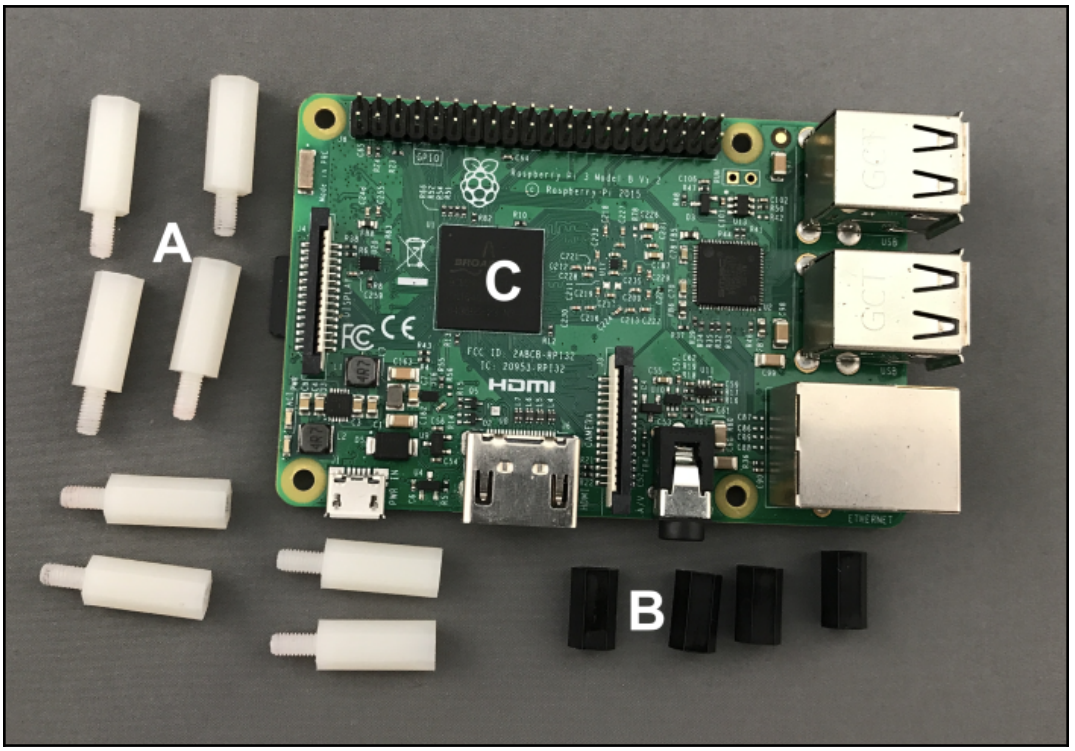


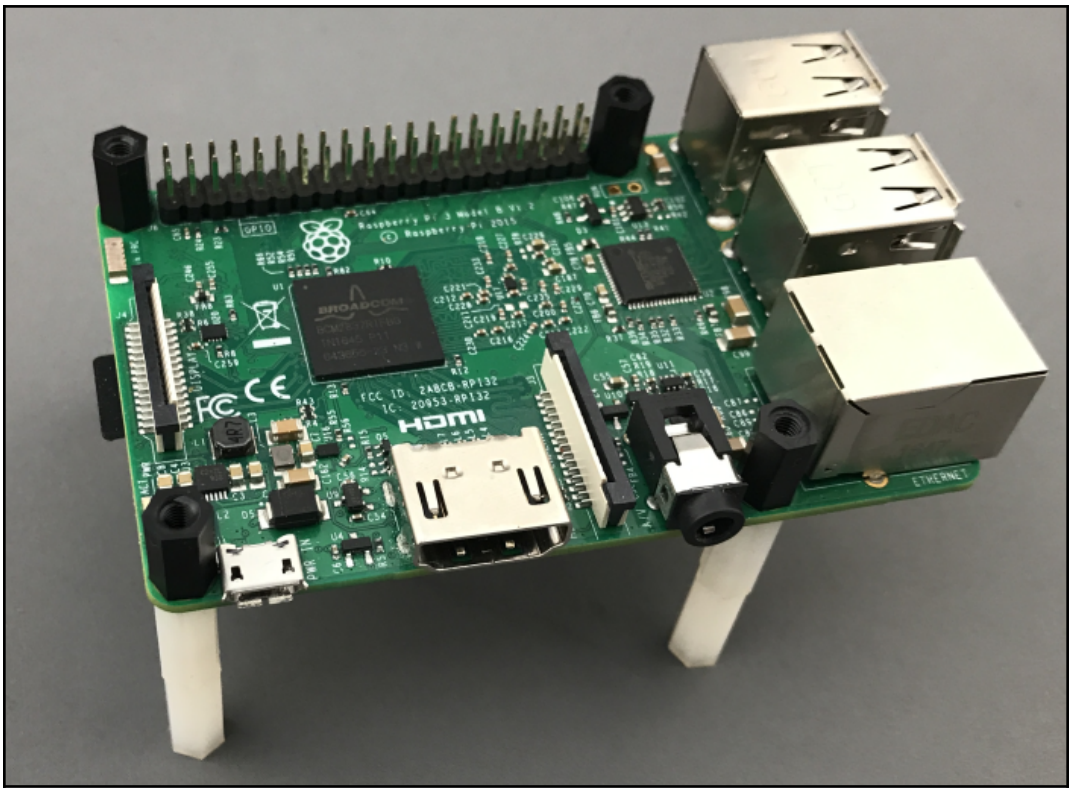

BUZZER

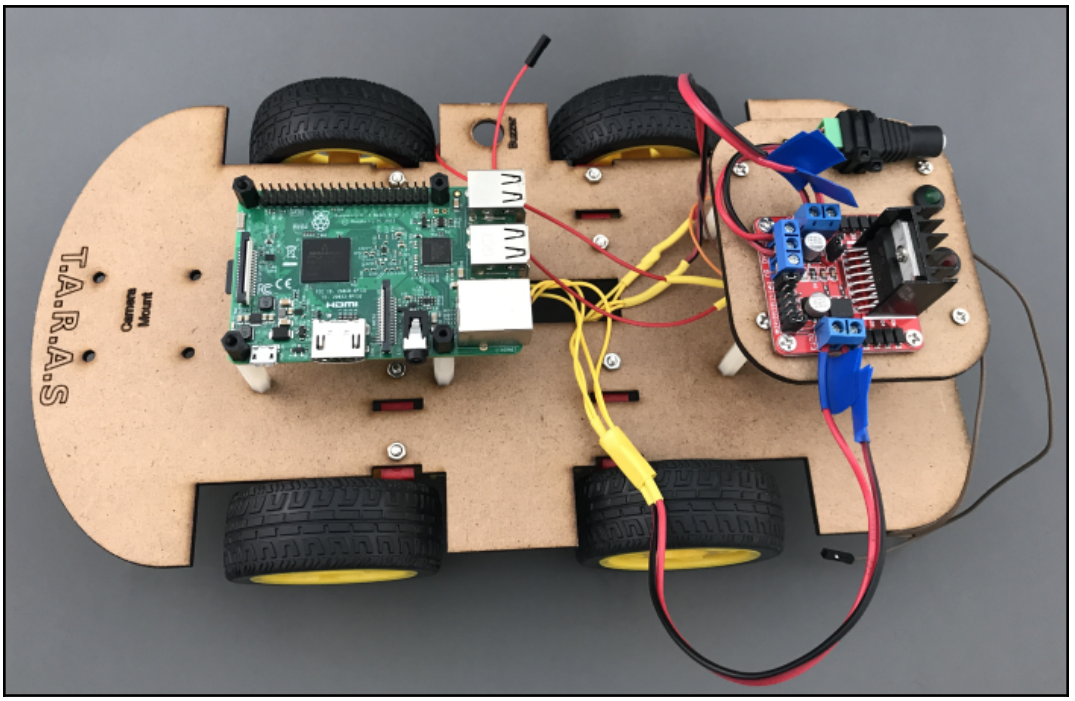

T.A.R.A.S
Camera Mount
Raspberry Pi and Servo Board
Motor Driver Board

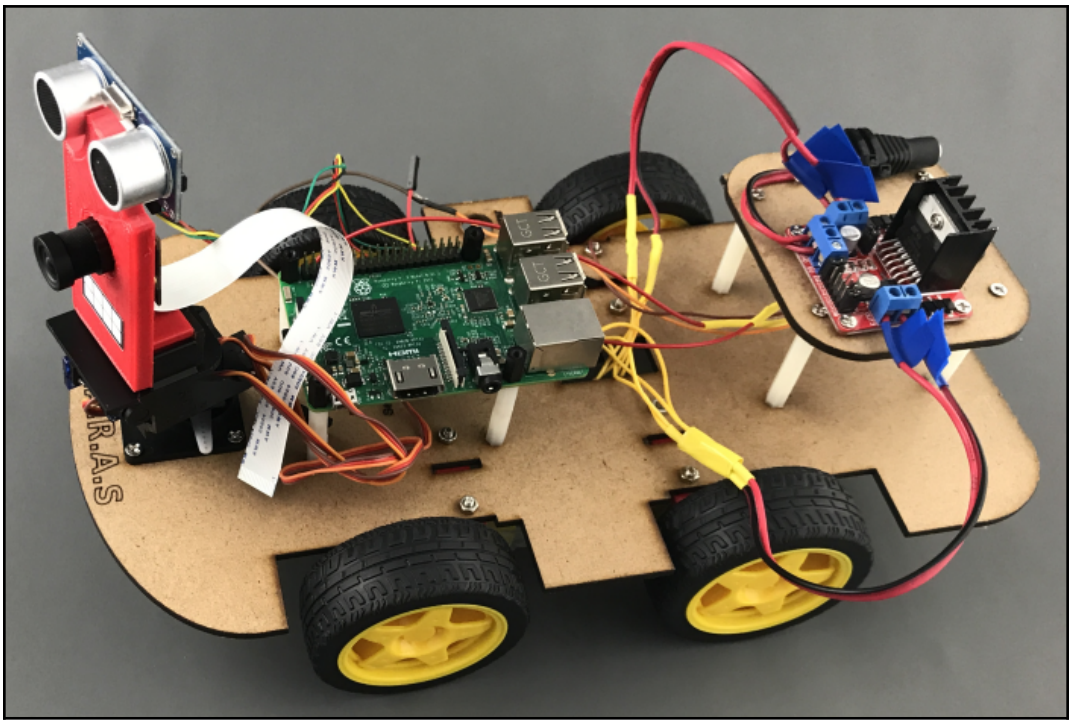

T.A.R.A.S
Camera Mount
Raspberry Pi and Servo Board
Buzzer

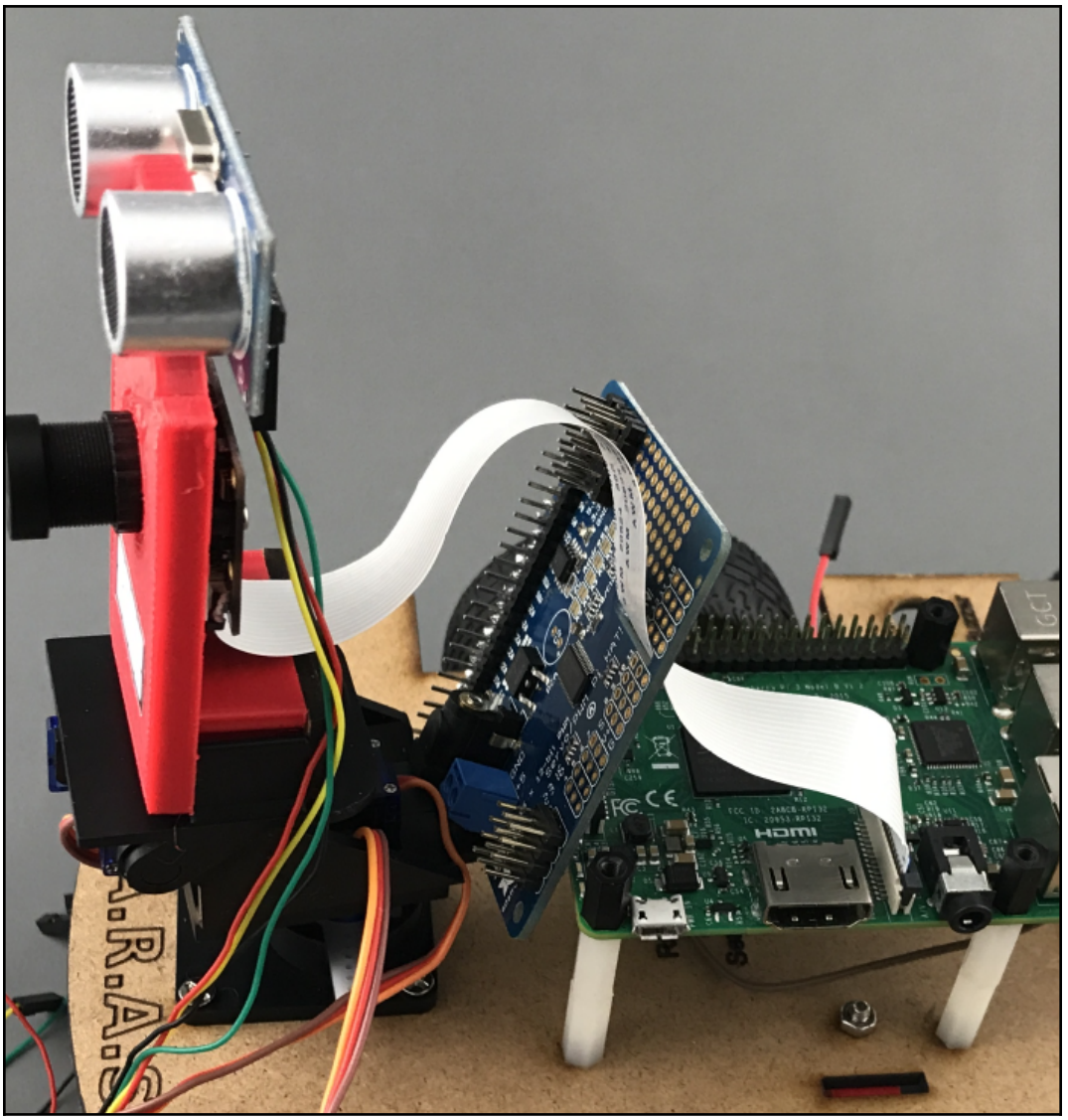

T.A.R.A.S
Camera Mount
Raspberry Pi and Servo Board
Buzzer

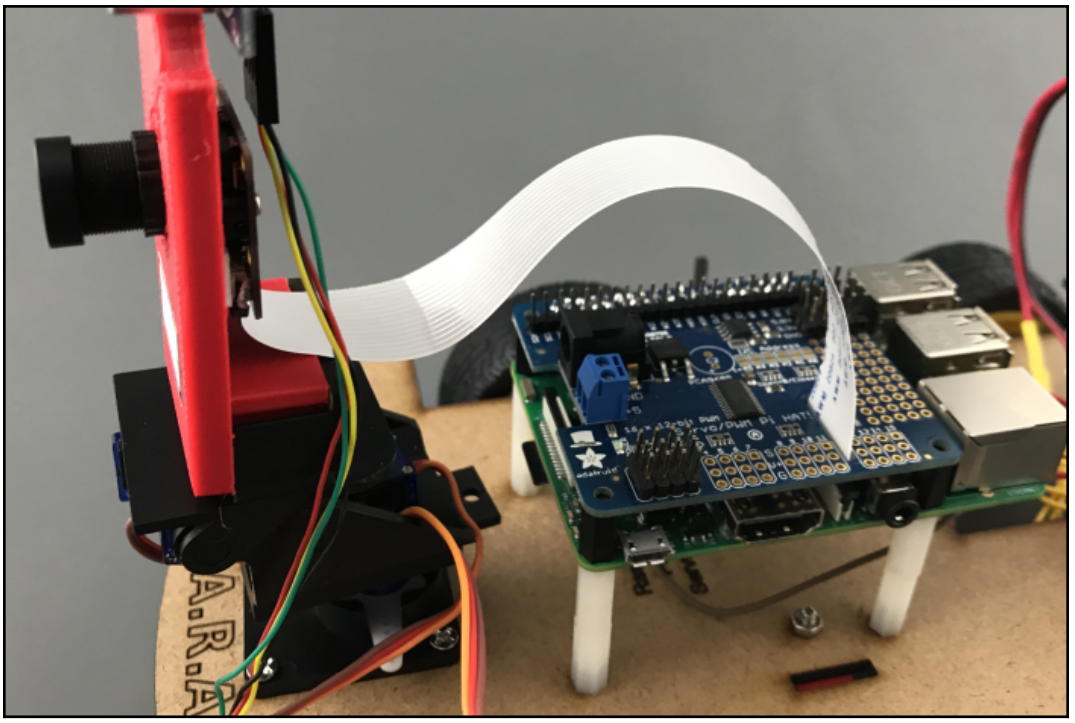

A
C
B
HDMI

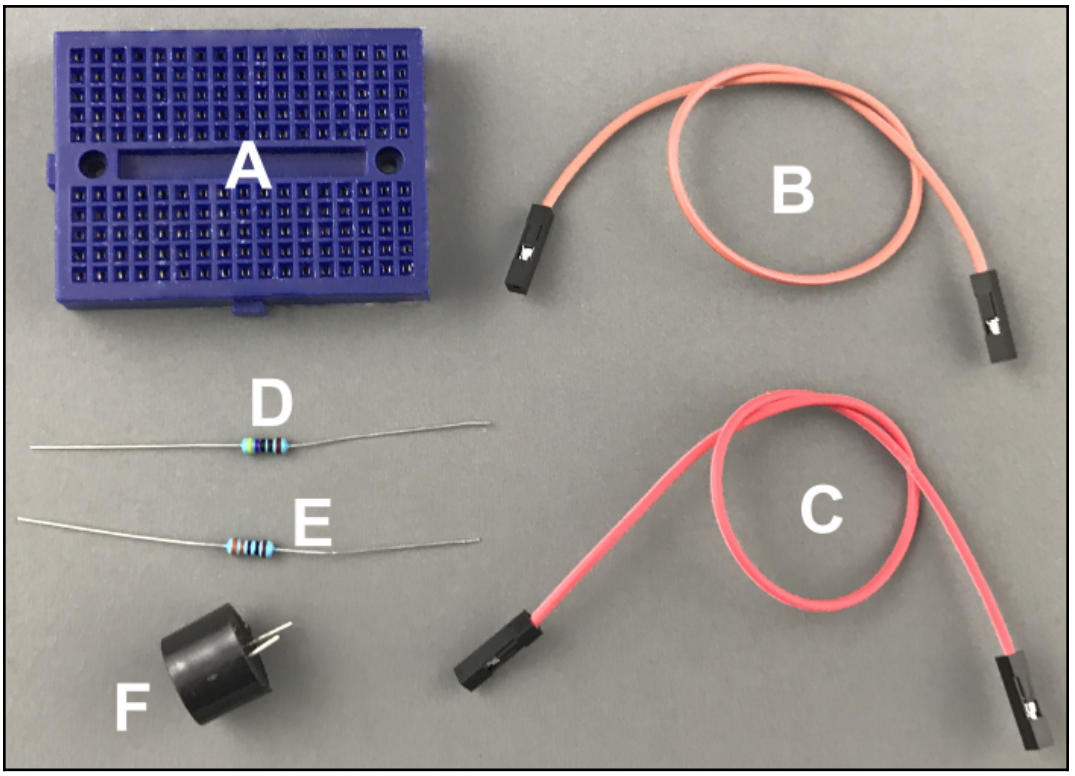

GPIO
BROADCOM
FCC ID: 2ABCB-RPI32
IC: 20953-RPI32
HDMI
ETHERNET

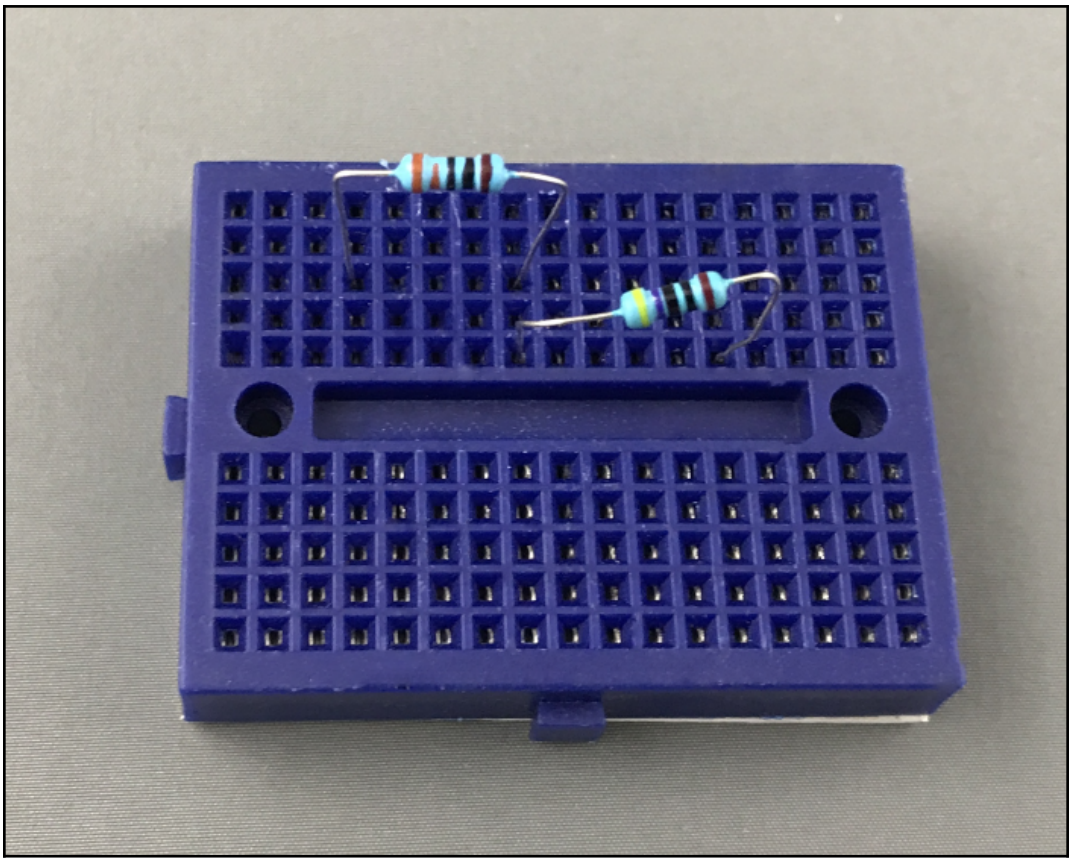

T.A.R.A.S
Camera
Mount

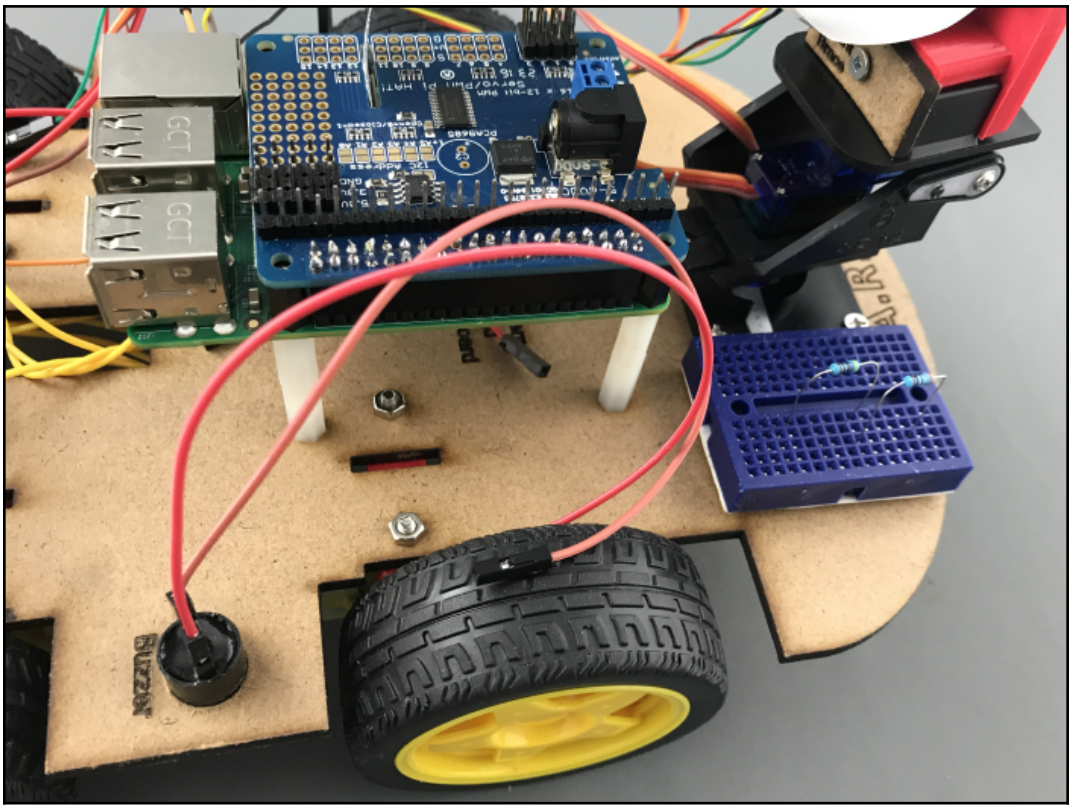

R.A.S

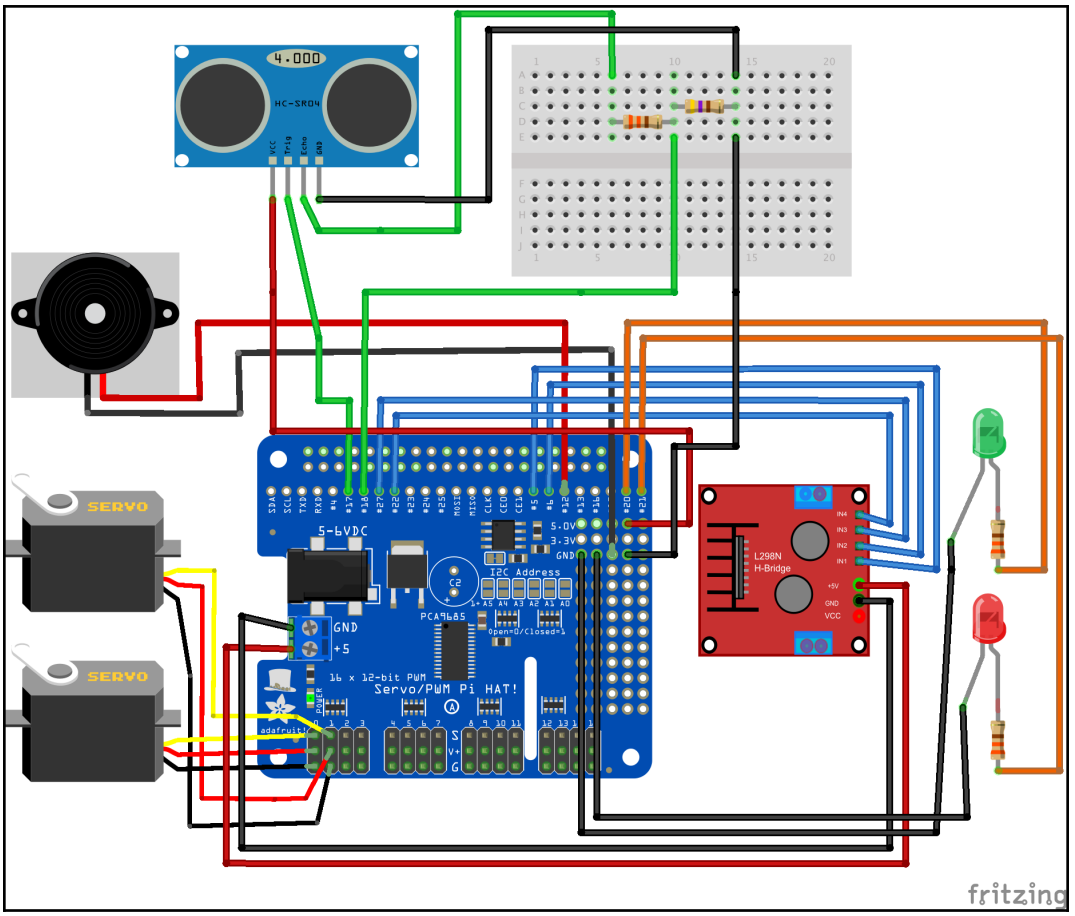

HDMI



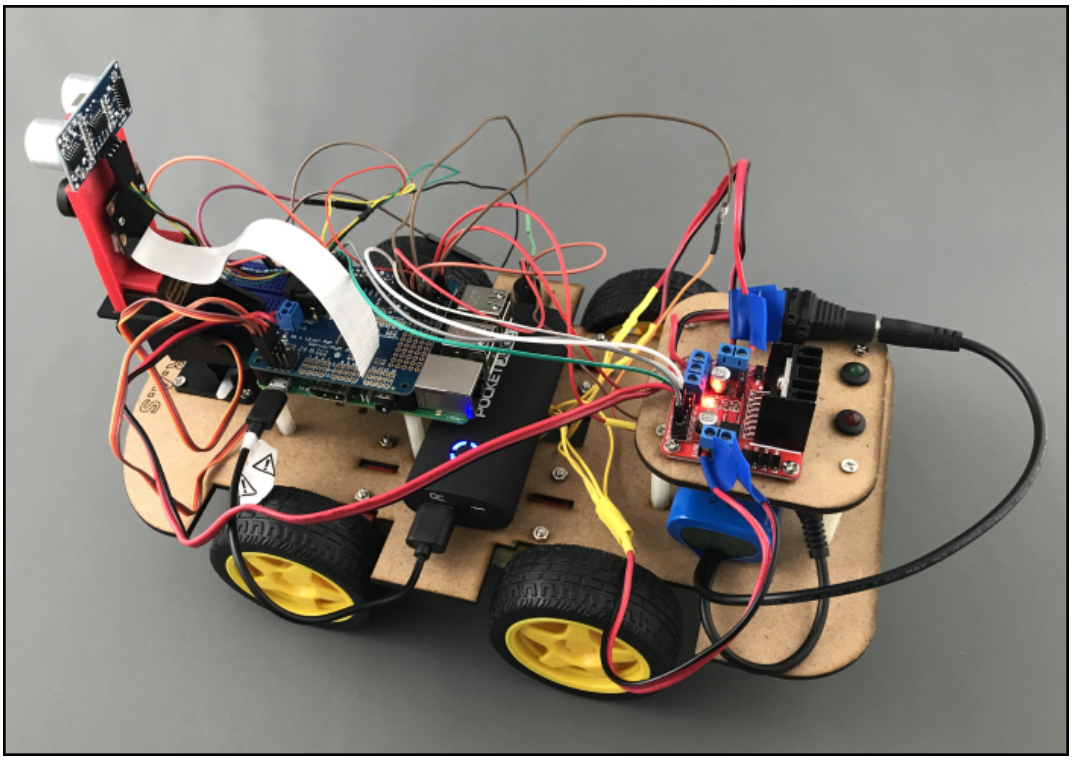

A
B
C
D
E
F

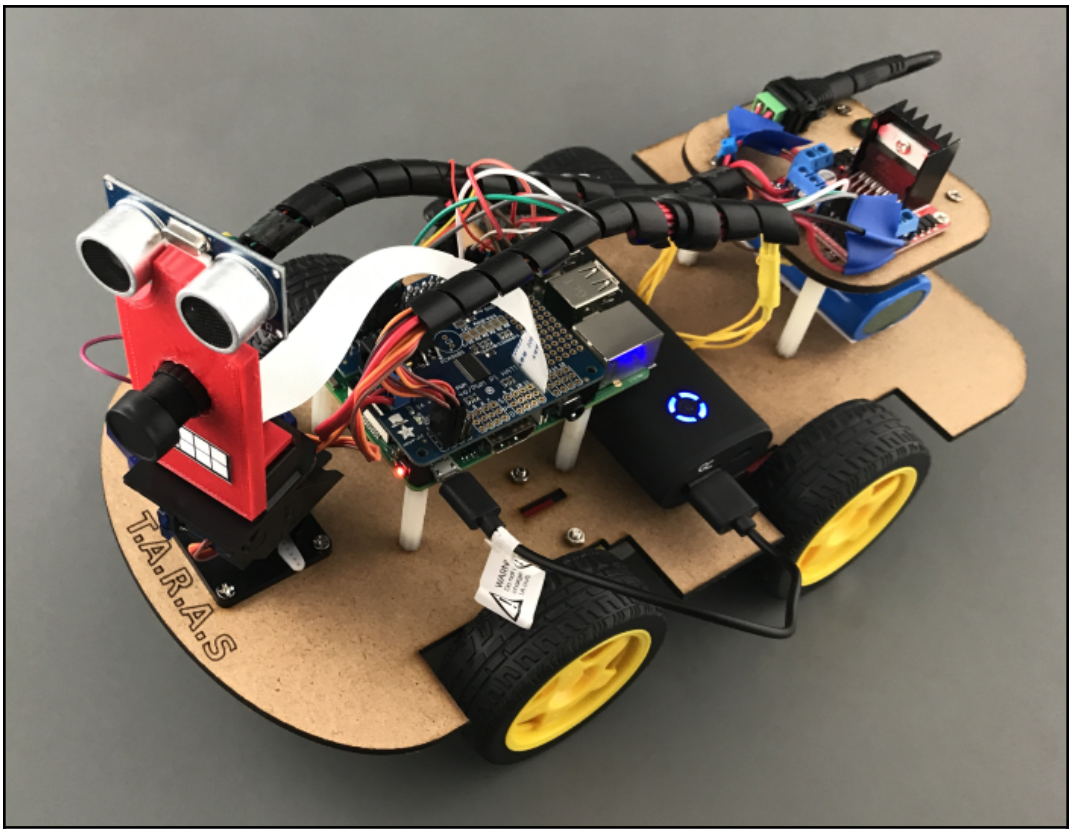


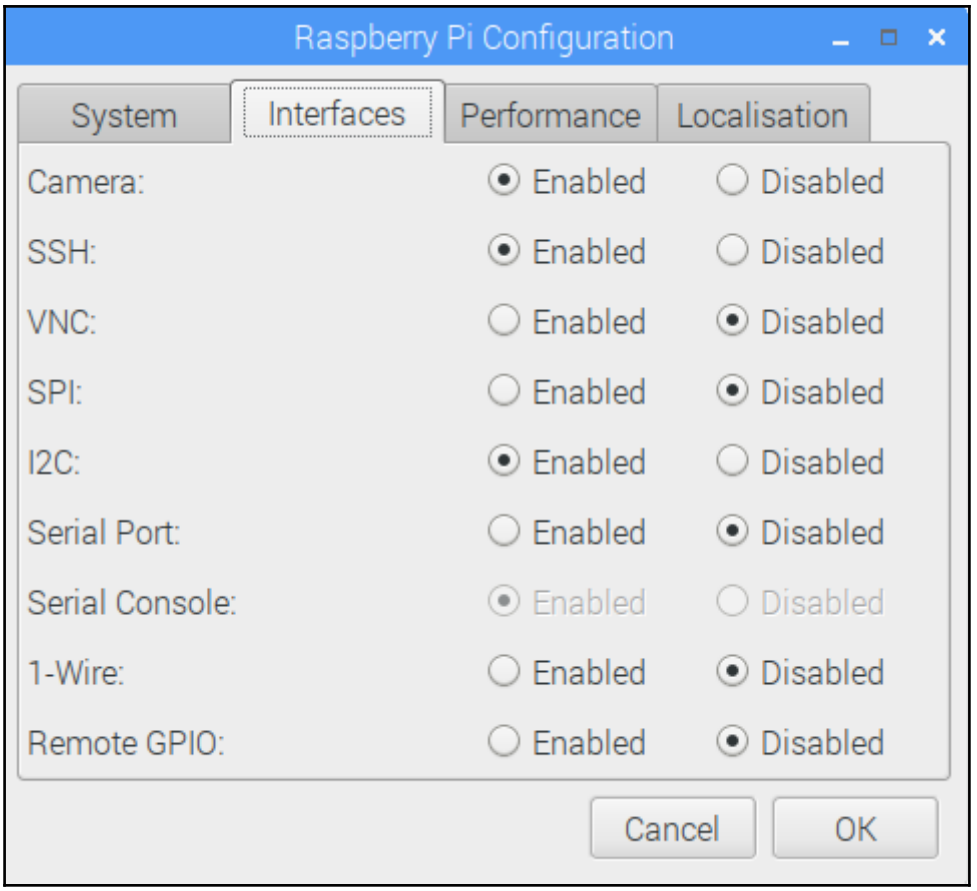


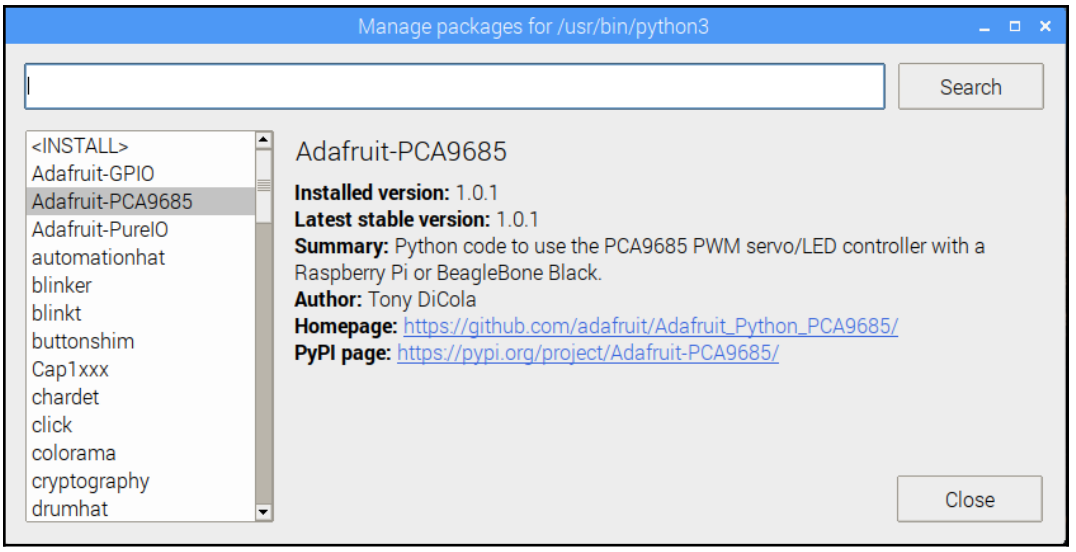

4.000
HC-SR04
VCC
Trig
Echo
GND
SERVO
SERVO
5-6VDC
GND
+5
5.0V
3.3V
GND
I2C Address
PCA9685
Open=0/Closed=1
16 x 12-bit PWM
Servo/PWM Pi HAT!
adafruit
L298N
H-Bridge
IN4
IN3
IN2
IN1
+5V
GND
VCC
fritzing


POCKETJUICE
Xiao-R Technology Co.,Ltd
www.wifi-robots.com
7.4V-2200mAh

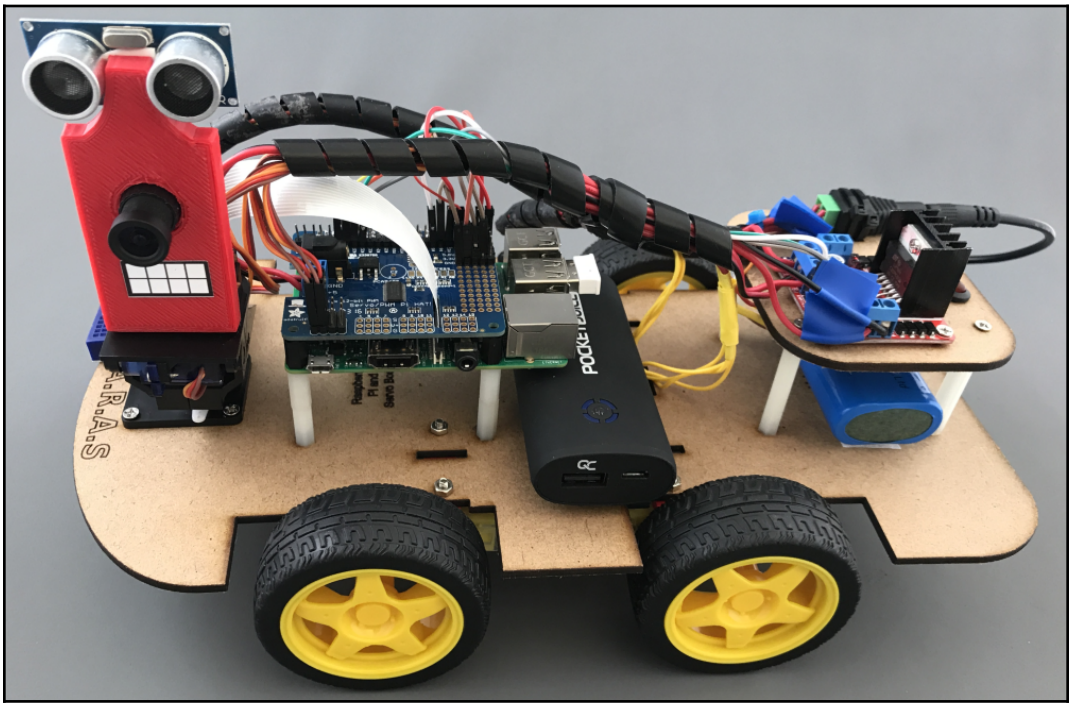


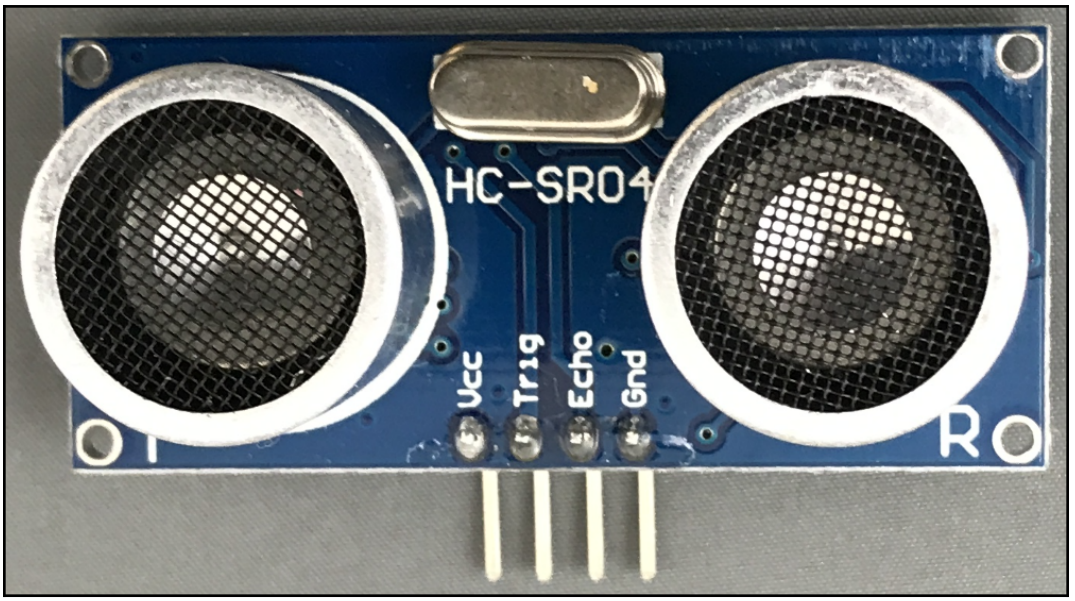

T.A.R.A.S

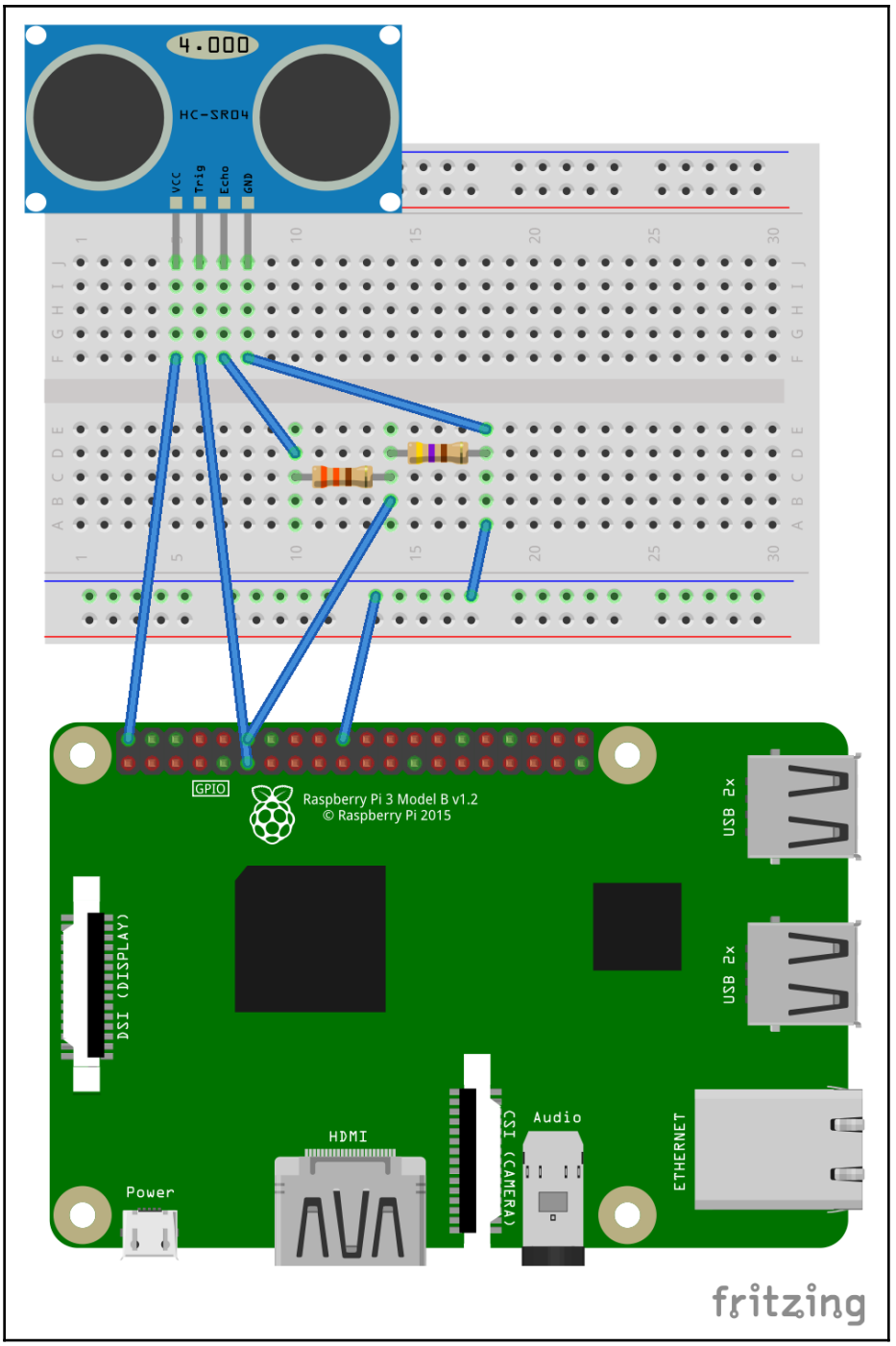

Raspberry Pi Configuration
System
Interfaces
Performance
Localisation
Camera: Enabled Disabled
SSH: Enabled Disabled
VNC: Enabled Disabled
SPI: Enabled Disabled
I2C: Enabled Disabled
Serial Port: Enabled Disabled
Serial Console: Enabled Disabled
1-Wire: Enabled Disabled
Remote GPIO: Enabled Disabled
Cancel
OK

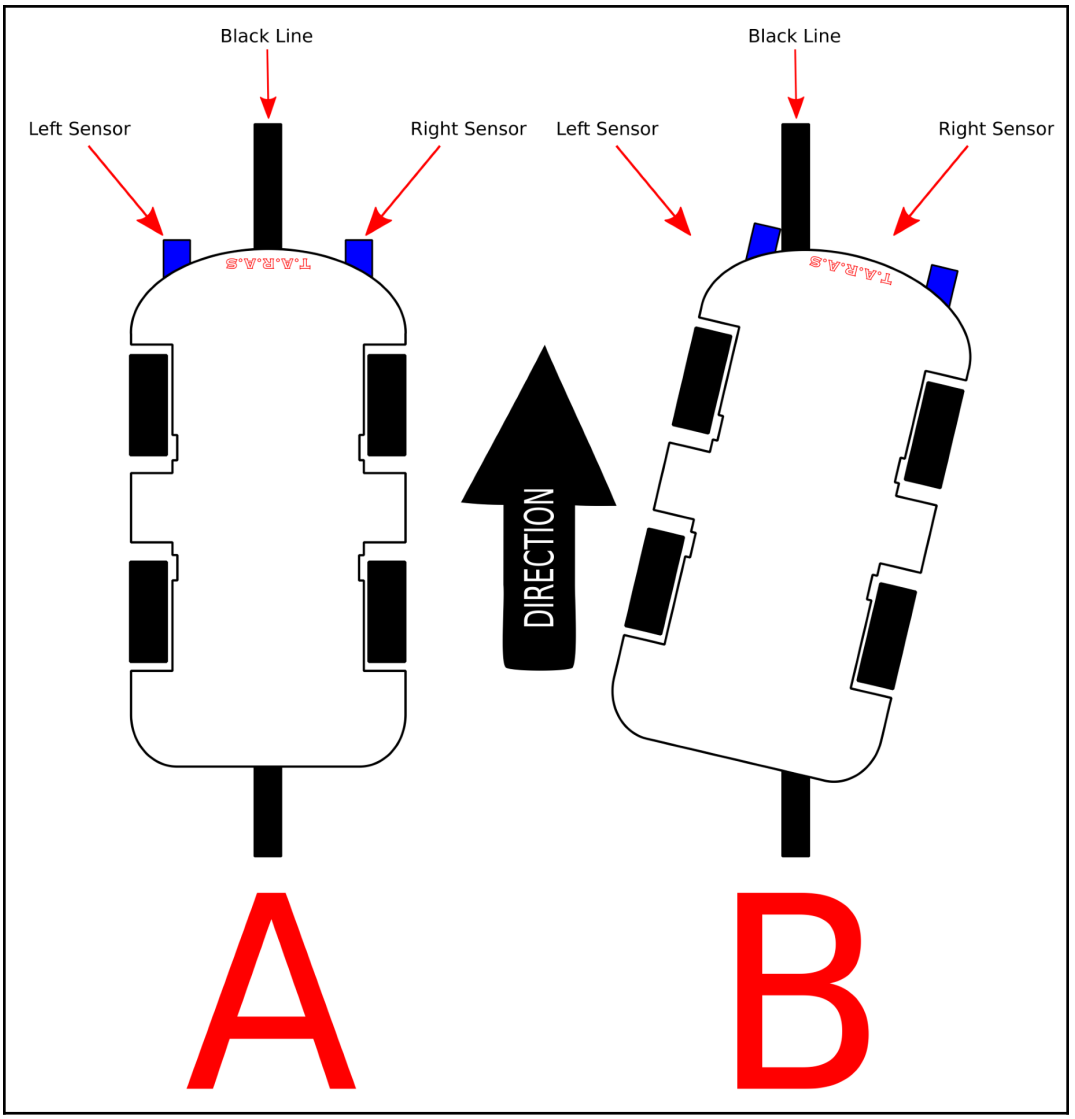

Manage packages for /usr/bin/python3
Search
<INSTALL>
Adafruit-GPIO
Adafruit-PCA9685
Adafruit-PureIO
automationhat
blinker
blinkt
buttonshim
Cap1xxx
chardet
click
colorama
cryptography
drumhat
Adafruit-PCA9685
Installed version: 1.0.1
Latest stable version: 1.0.1
Summary: Python code to use the PCA9685 PWM servo/LED controller with a Raspberry Pi or BeagleBone Black.
Author: Tony DiCola
Homepage: https://github.com/adafruit/Adafruit_Python_PCA9685/
PyPI page: https://pypi.org/project/Adafruit-PCA9685/
Close

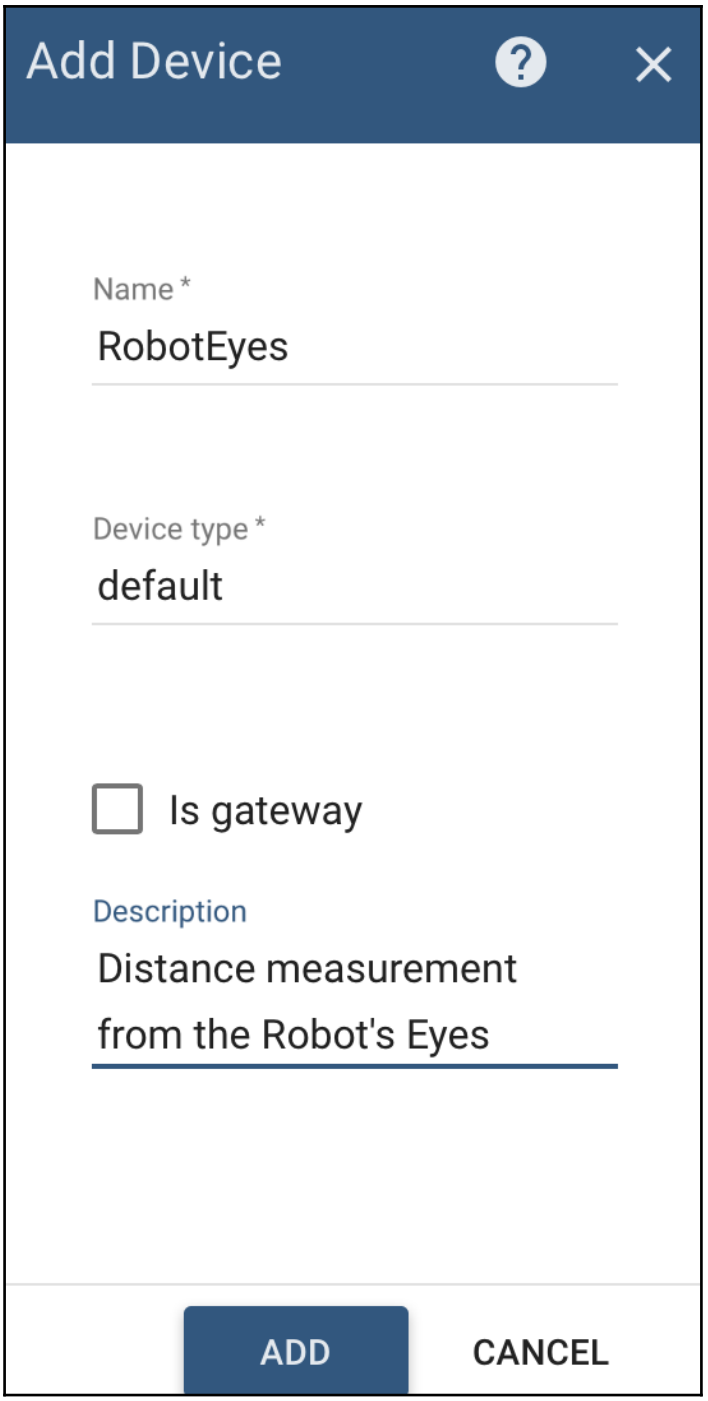
Chapter 24: Connecting Sensory Inputs from the Robot Car to the Web

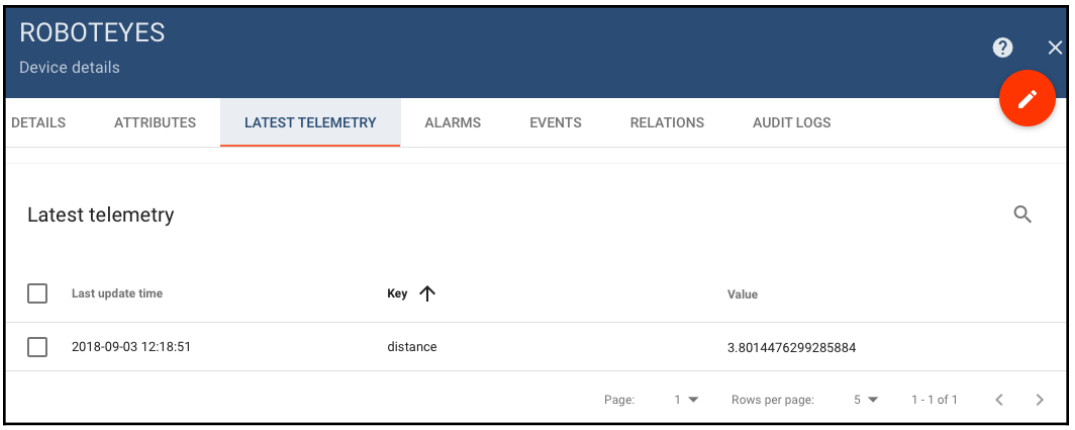



HC-SR04
Vcc
Trig
Echo
Gnd
R


4.000
HC-SR04
VCC
Trig
Echo
GND
GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
CSI (CAMERA)
Audio
ETHERNET
HDMI
Power
fritzing

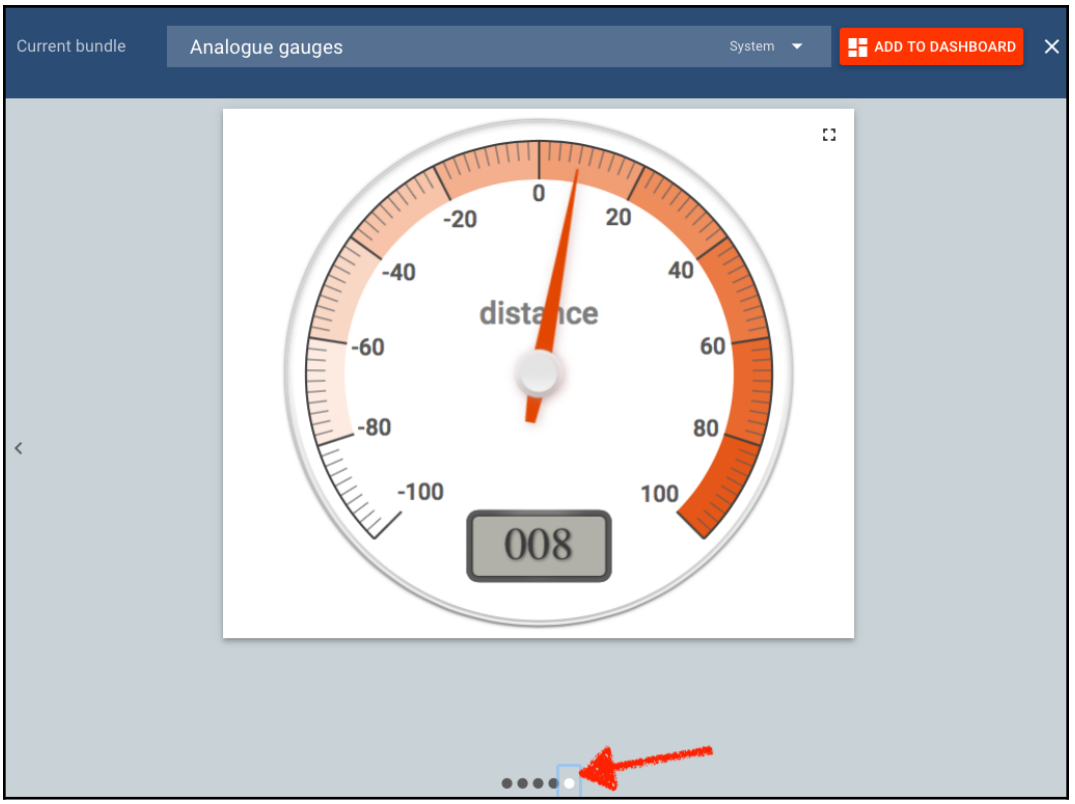

Black Line
Left Sensor
Right Sensor
T.A.R.A.S
DIRECTION
Black Line
Left Sensor
Right Sensor
T.A.R.A.S
A
B

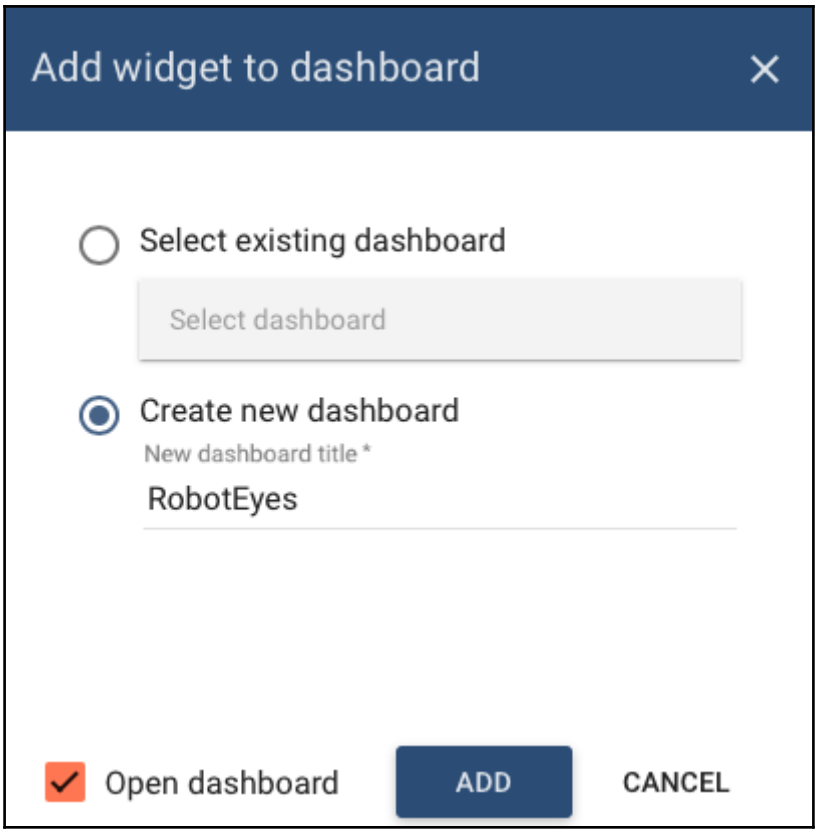

Add Device
Name *
RobotEyes
Device type *
default
Is gateway
Description
Distance measurement
from the Robot's Eyes
ADD
CANCEL


ROBOTEYES
Device details
DETAILS
ATTRIBUTES
LATEST TELEMETRY
ALARMS
EVENTS
RELATIONS
AUDIT LOGS
Latest telemetry
Last update time
Key
Value
2018-09-03 12:18:51
distance
3.8014476299285884
Page: 1
Rows per page: 5
1 - 1 of 1


1 telemetry unit selected
SHOW ON WIDGET
Last update time
Key
Value
2018-09-03 12:25:37
distance
7.5926653146743766
Page: 1
Rows per page: 5
1 - 1 of 1

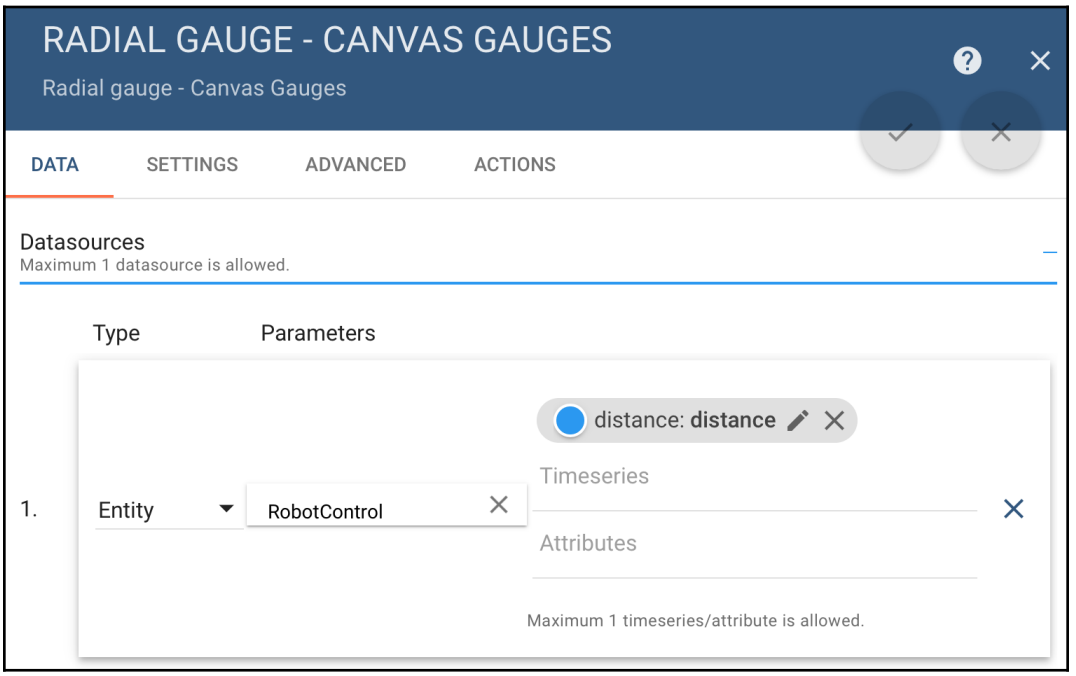

Current bundle
Analogue gauges
System
ADD TO DASHBOARD

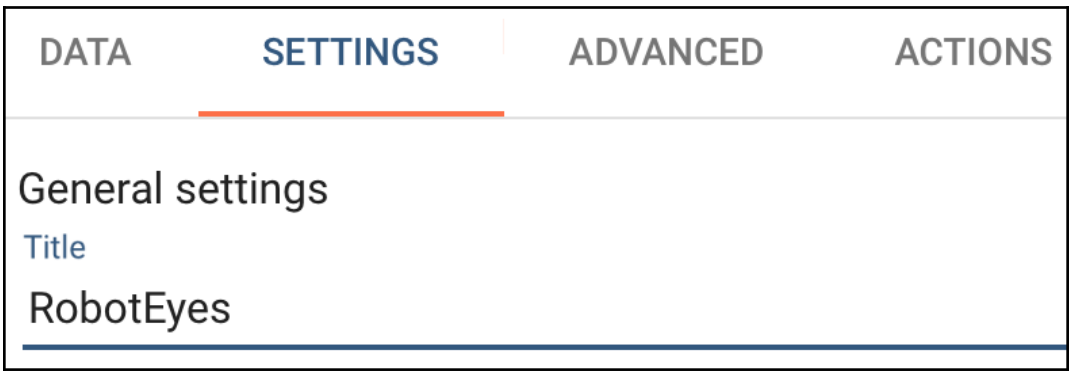

Current bundle
Analogue gauges
System
ADD TO DASHBOARD
0
-20
20
-40
40
distance
-60
60
-80
80
-100
100
008

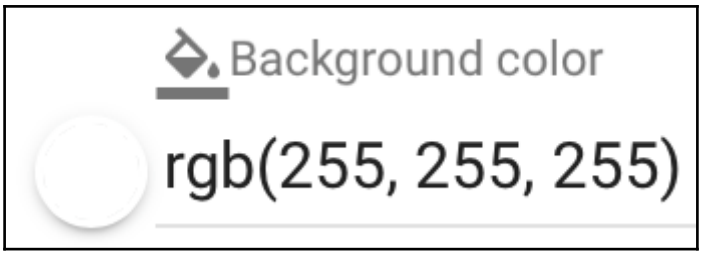

Add widget to dashboard
Select existing dashboard
Select dashboard
Create new dashboard
New dashboard title *
RobotEyes
Open dashboard
ADD
CANCEL

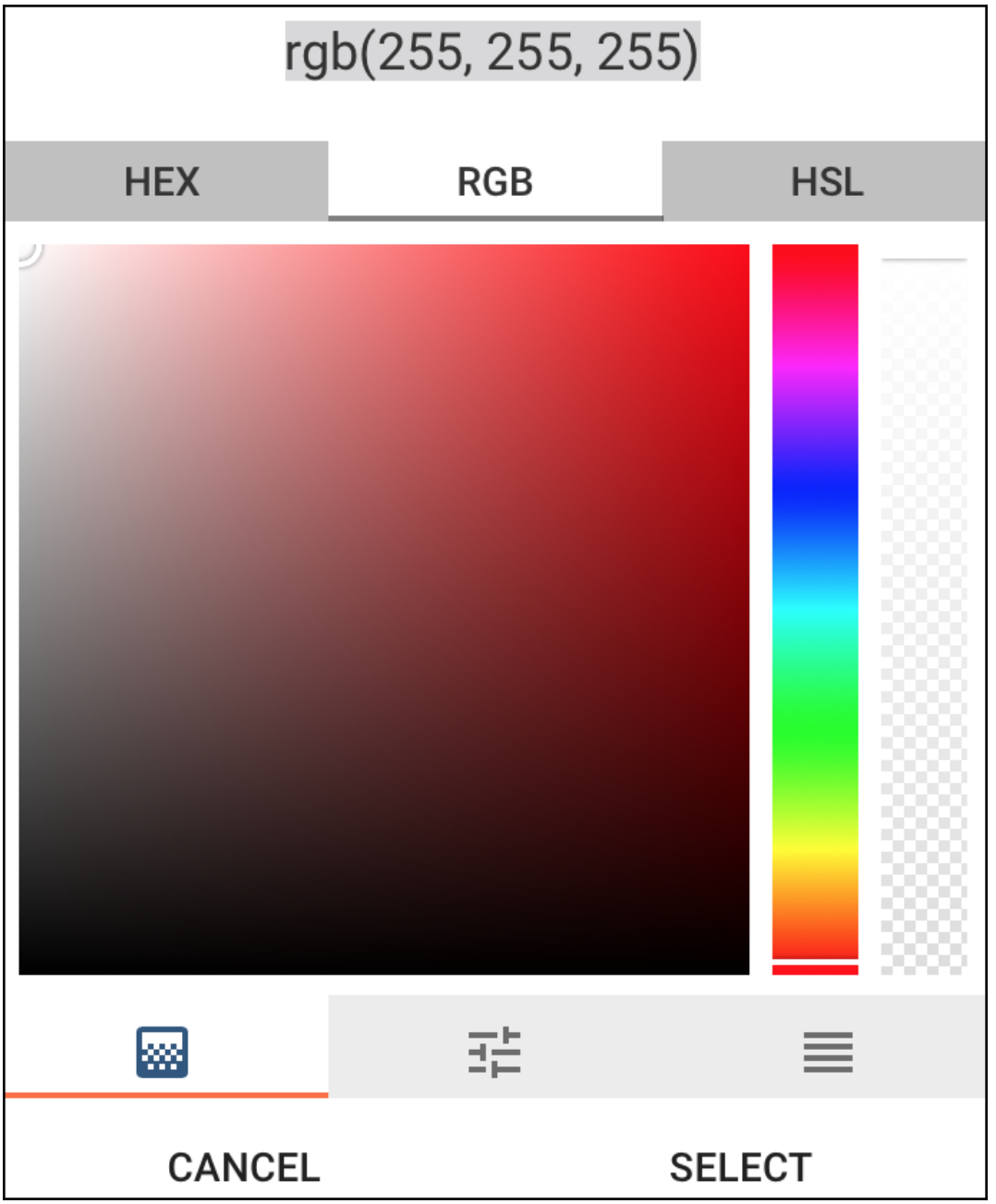

0
-20
20
-40
40
distance
-60
60
-80
80
-100
100
006

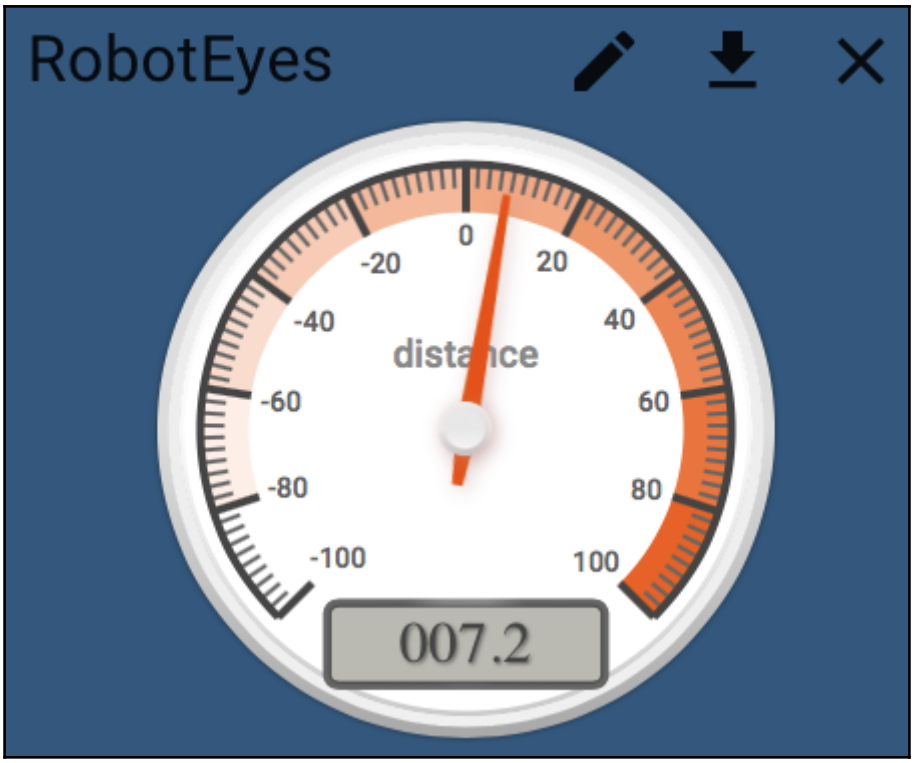
Chapter 25: Controlling the Robot Car with Web Service Calls

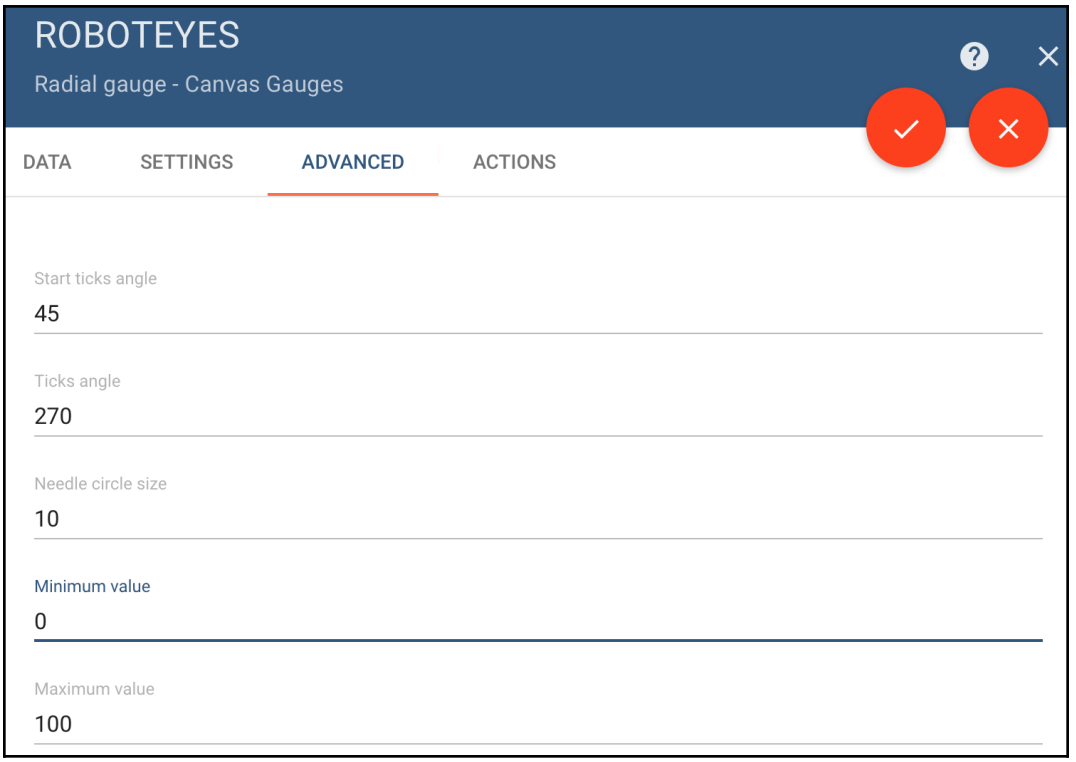


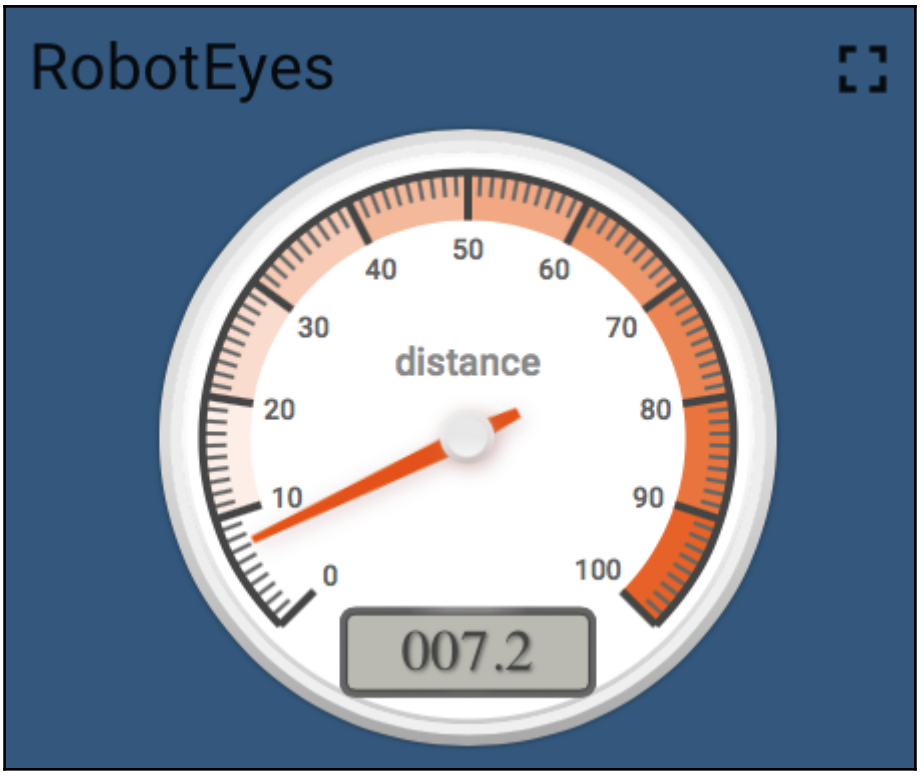


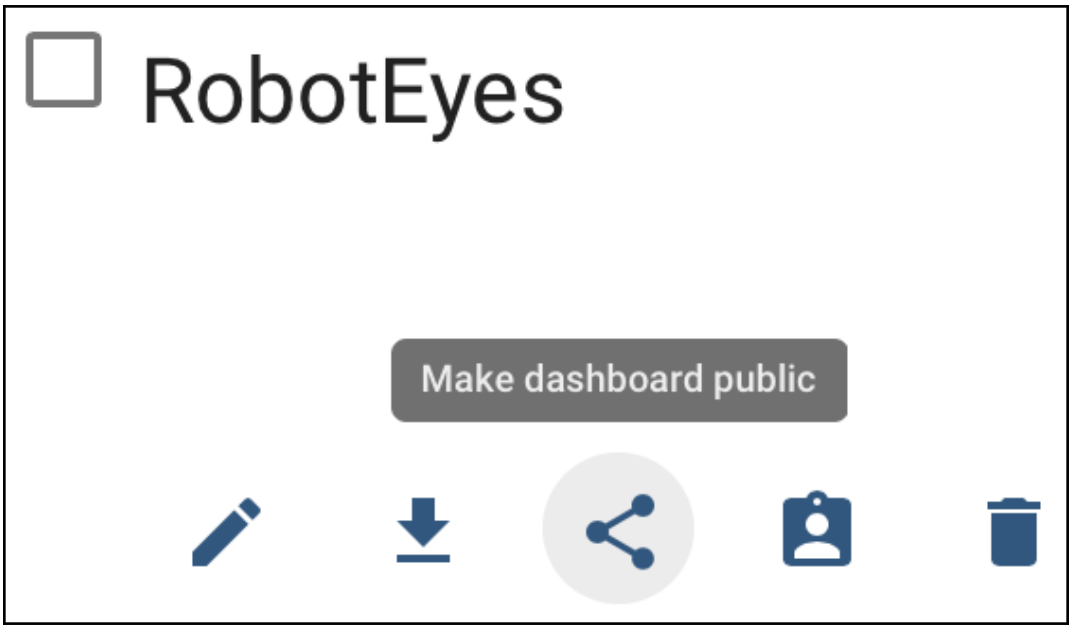

Background color
rgb(255, 255, 255)

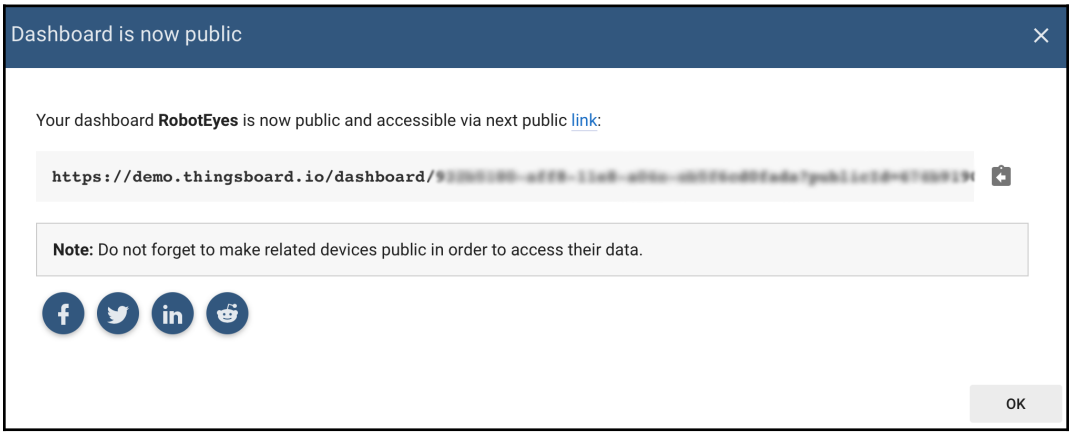

rgb(255, 255, 255)
HEX
RGB
HSL
CANCEL
SELECT

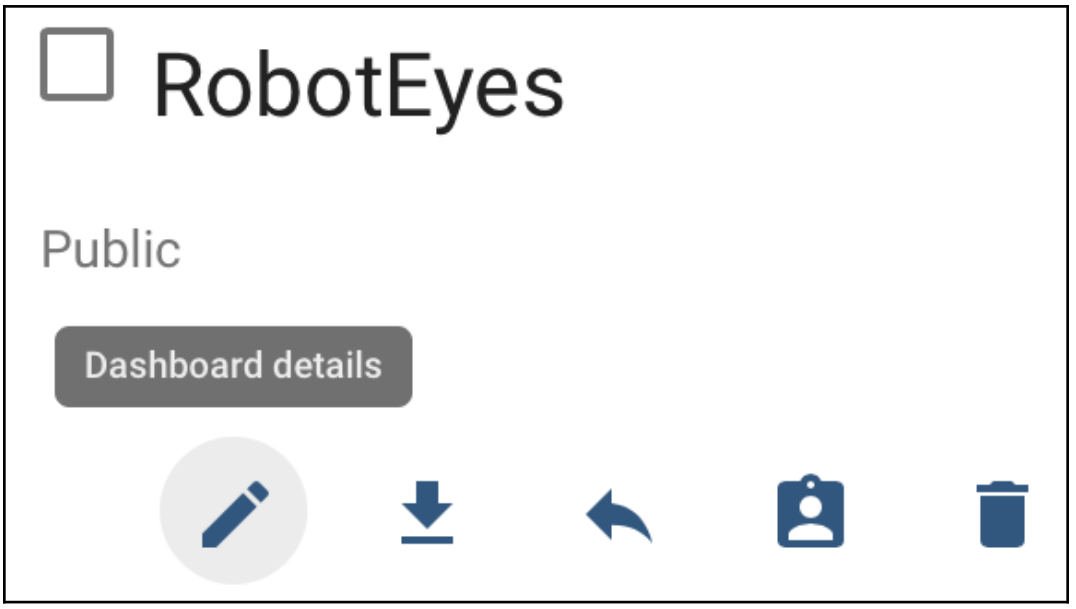

RobotEyes
0
-20
20
-40
40
distance
-60
60
-80
80
-100
100
007.2

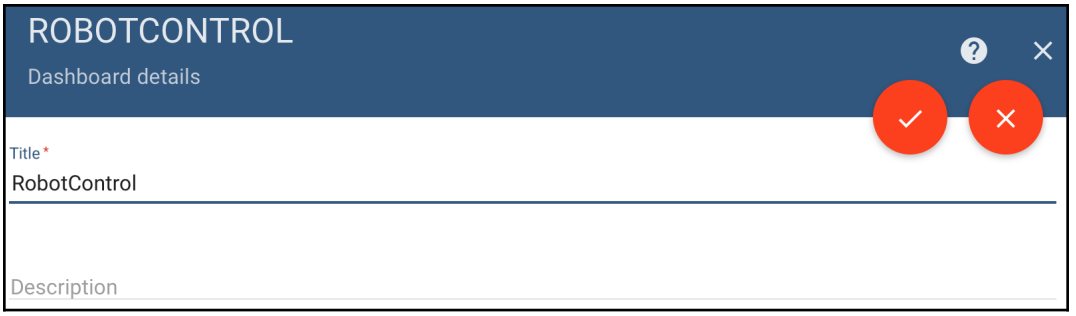

ROBOTEYES
Radial gauge - Canvas Gauges
DATA
SETTINGS
ADVANCED
ACTIONS
Start ticks angle
45
Ticks angle
270
Needle circle size
10
Minimum value
0
Maximum value
100


RobotEyes
0
10
20
30
40
50
60
70
80
90
100
distance
007.2

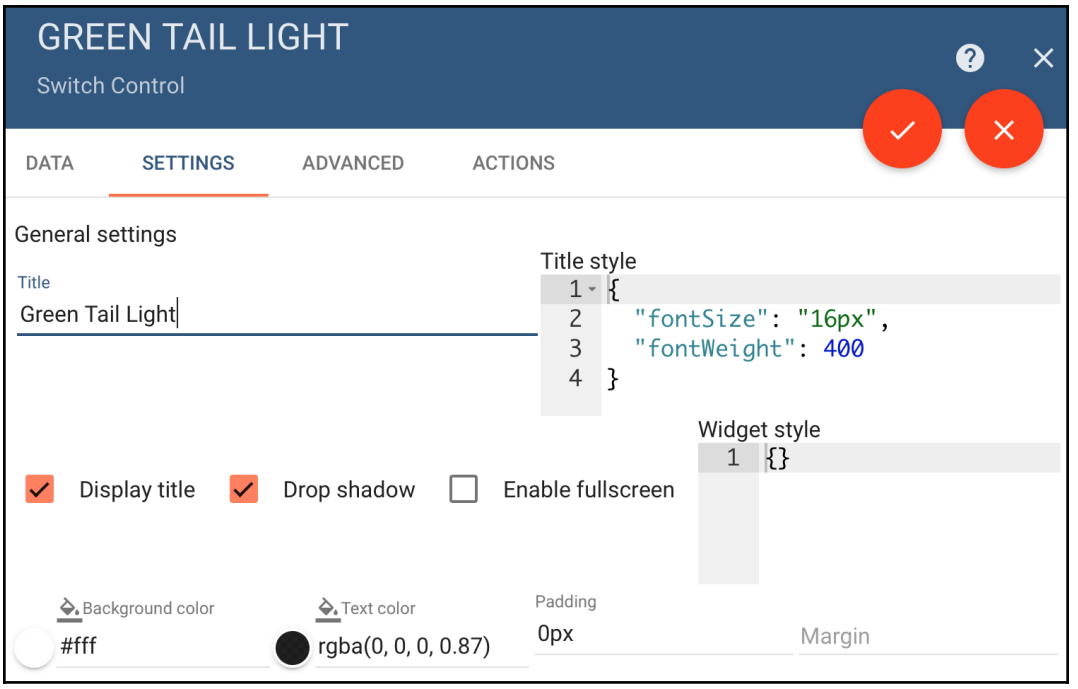

RobotEyes
Make dashboard public

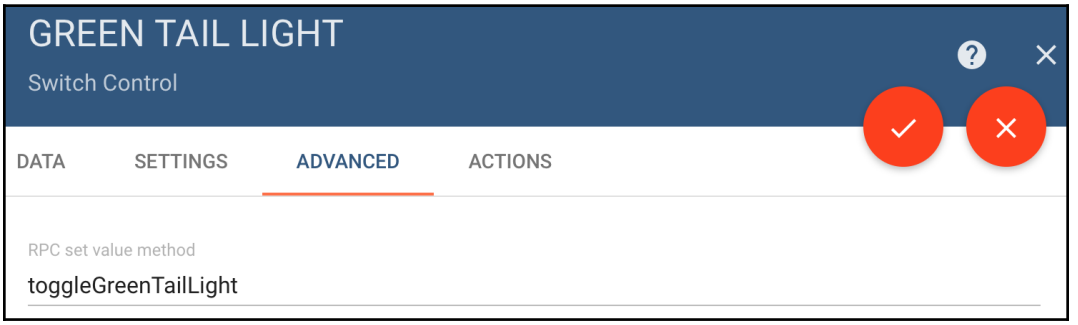

Dashboard is now public
Your dashboard RobotEyes is now public and accessible via next public link:
https://demo.thingsboard.io/dashboard/
Note: Do not forget to make related devices public in order to access their data.
OK


RobotEyes
Public
Dashboard details

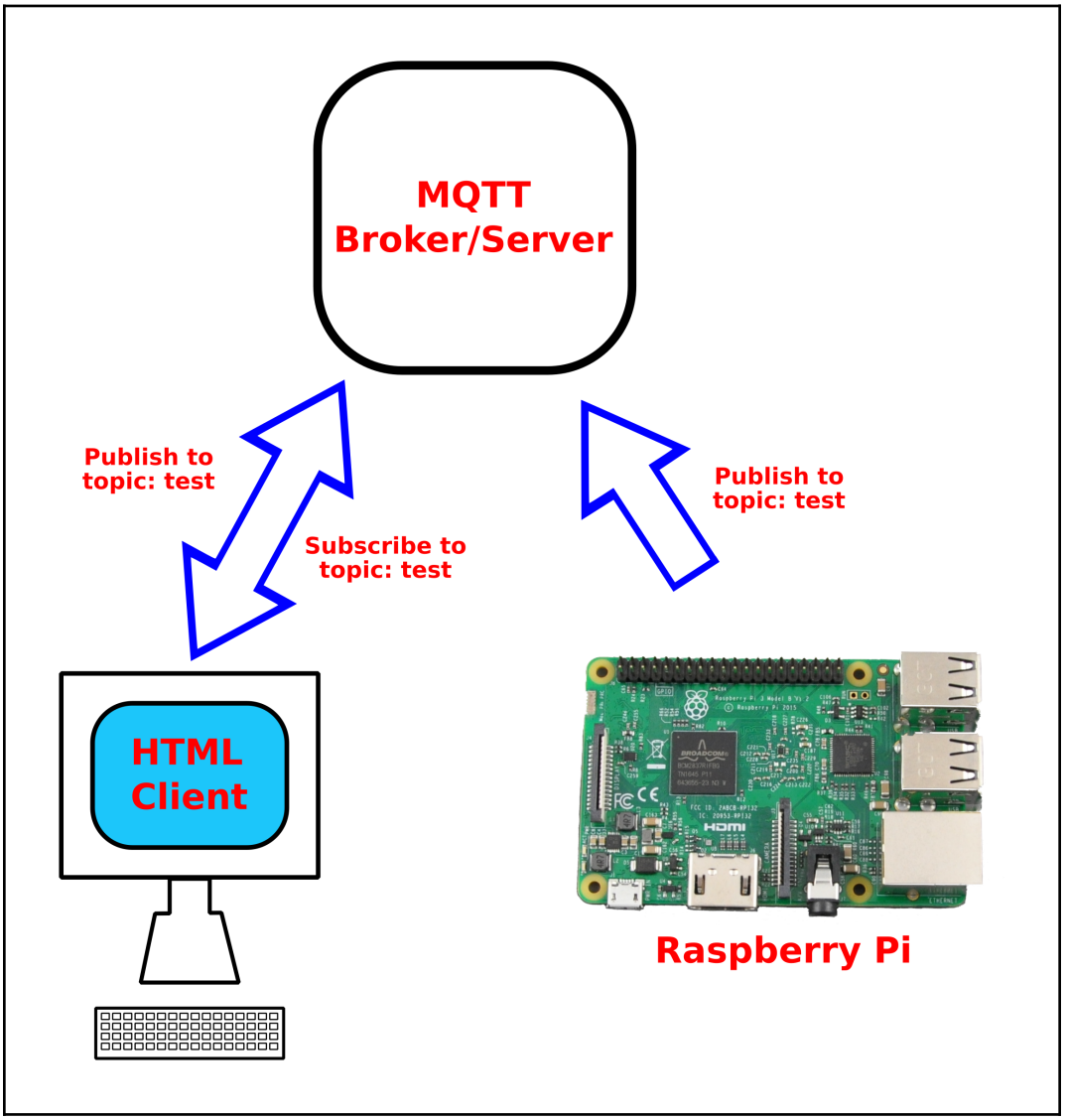

ROBOTCONTROL
Dashboard details
Title *
RobotControl
Description

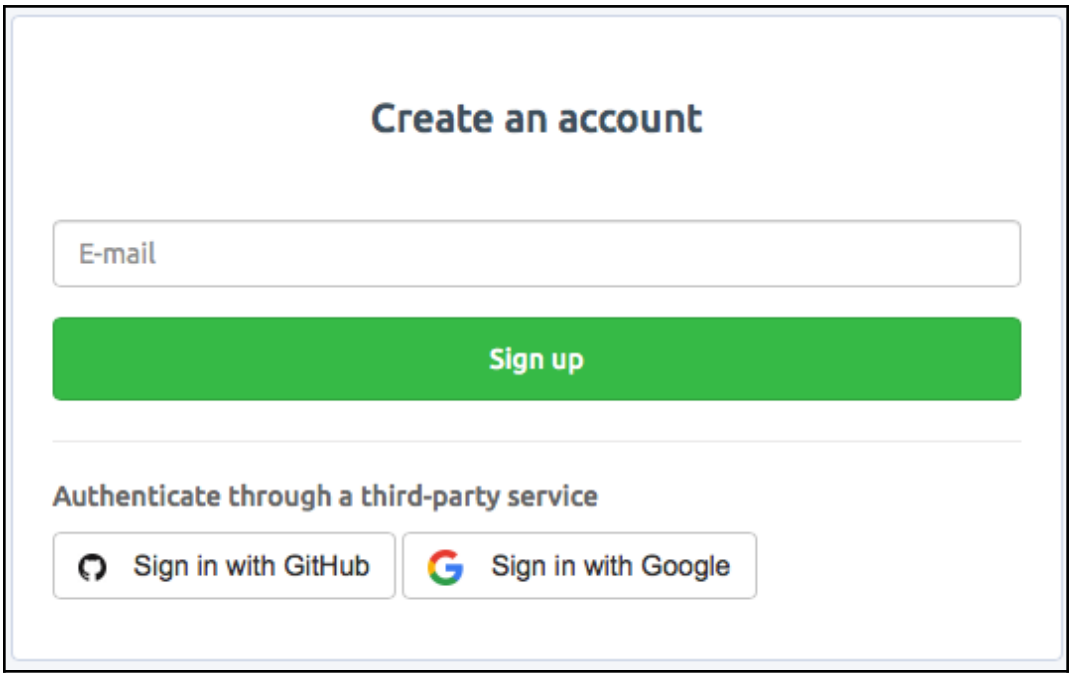

SELECT WIDGET
Current bundle
Control widgets
System
CONTROL WIDGET
Target device is not set!
Led indicator
Target device is not set!
Round switch
I
0
Target device is not set!
Switch control
OFF

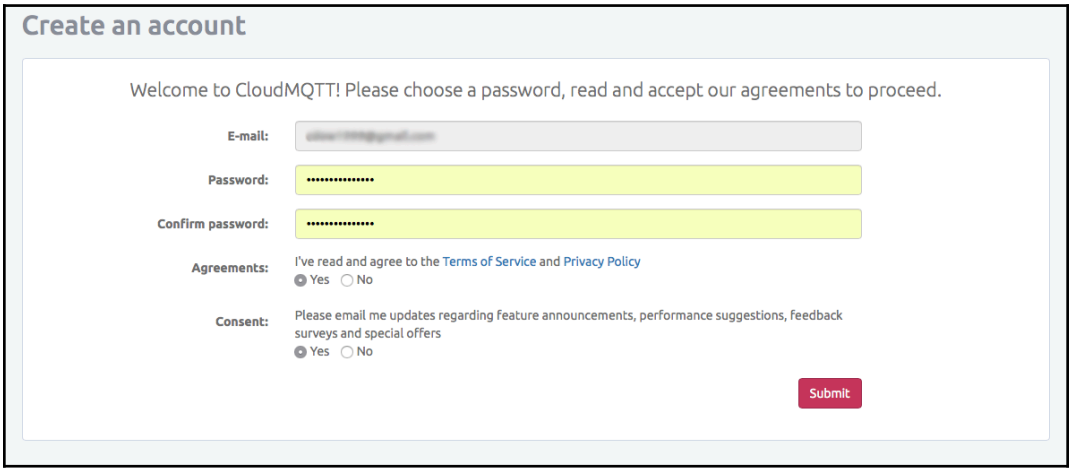

GREEN TAIL LIGHT
Switch Control
DATA
SETTINGS
ADVANCED
ACTIONS
General settings
Title
Green Tail Light
Title style
{
"fontSize": "16px",
"fontWeight": 400
}
Widget style
{}
Display title
Drop shadow
Enable fullscreen
Background color
#fff
Text color
rgba(0, 0, 0, 0.87)
Padding
0px
Margin

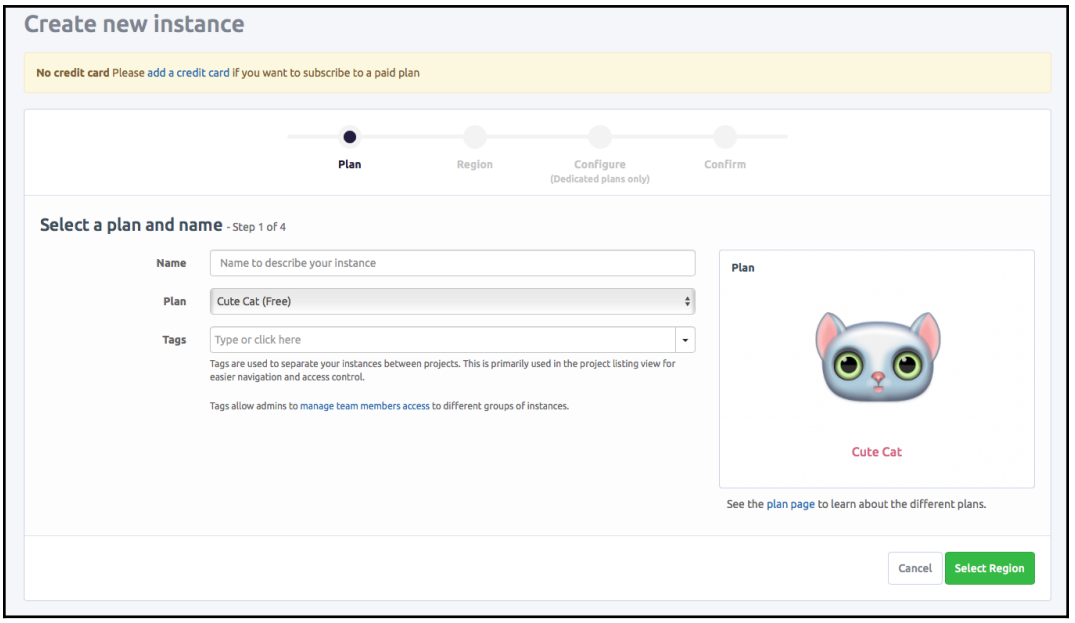

GREEN TAIL LIGHT
Switch Control
DATA
SETTINGS
ADVANCED
ACTIONS
RPC set value method
toggleGreenTailLight

Chapter 26: Building the JavaScript Client

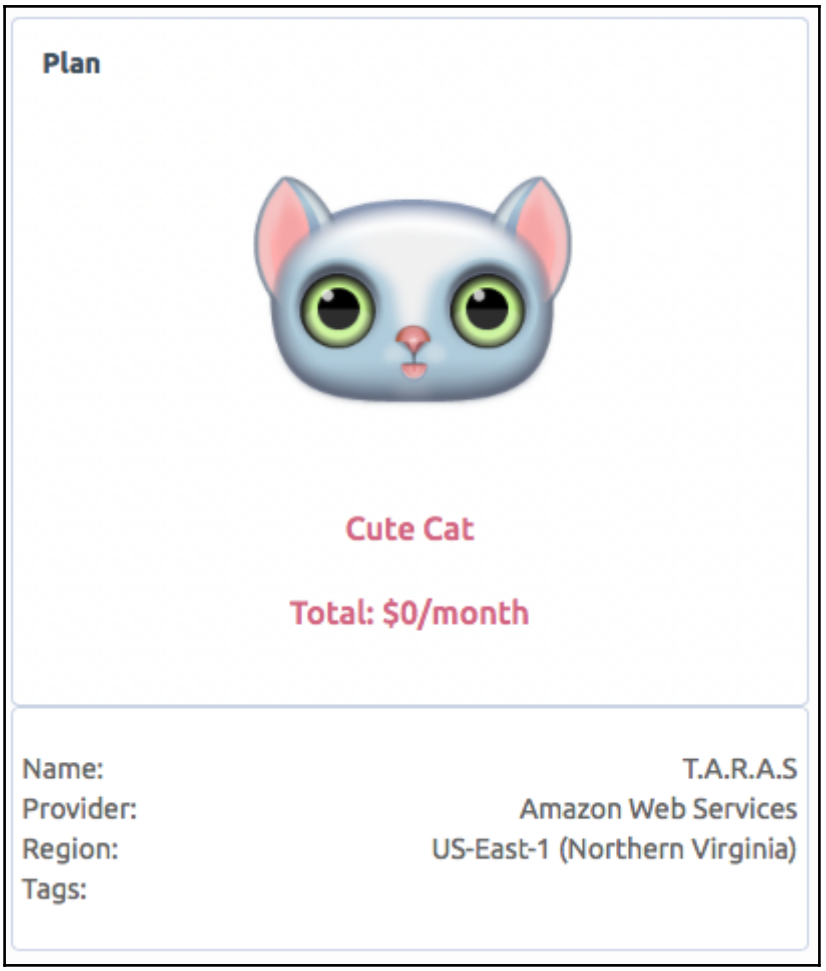



Create an account
E-mail
Sign up
Authenticate through a third-party service
Sign in with GitHub
Sign in with Google

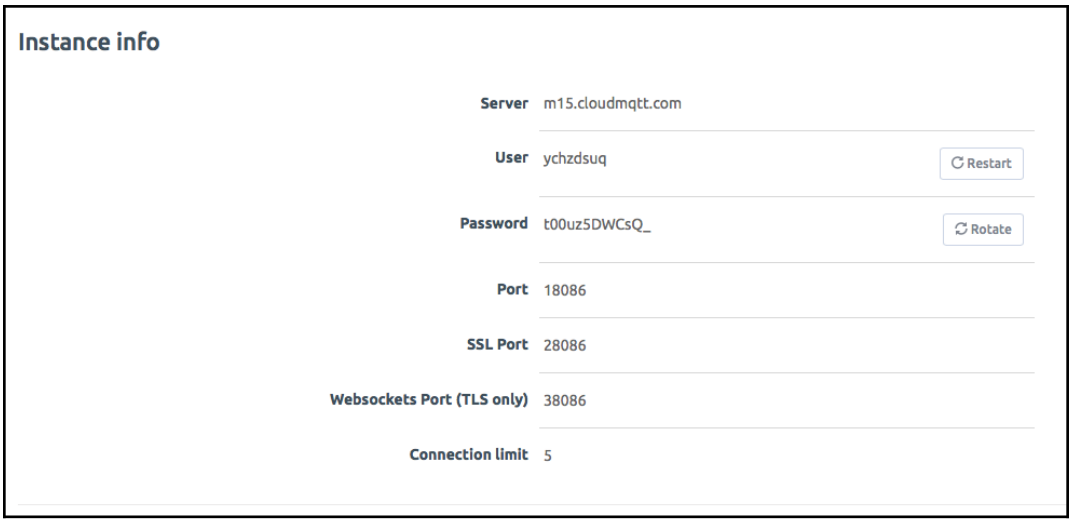

Create an account
Welcome to CloudMQTT! Please choose a password, read and accept our agreements to proceed.
E-mail:
Password:
Confirm password:
Agreements:
I've read and agree to the Terms of Service and Privacy Policy
Yes
No
Consent:
Please email me updates regarding feature announcements, performance suggestions, feedback surveys and special offers
Yes
No
Submit

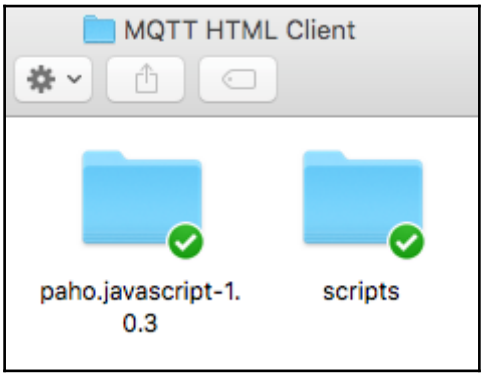

Create new instance
No credit card Please add a credit card if you want to subscribe to a paid plan
Plan
Region
Configure
(Dedicated plans only)
Confirm
Select a plan and name - Step 1 of 4
Name
Name to describe your instance
Plan
Cute Cat (Free)
Tags
Type or click here
Tags are used to separate your instances between projects. This is primarily used in the project listing view for easier navigation and access control.
Tags allow admins to manage team members access to different groups of instances.
Plan
Cute Cat
See the plan page to learn about the different plans.
Cancel
Select Region

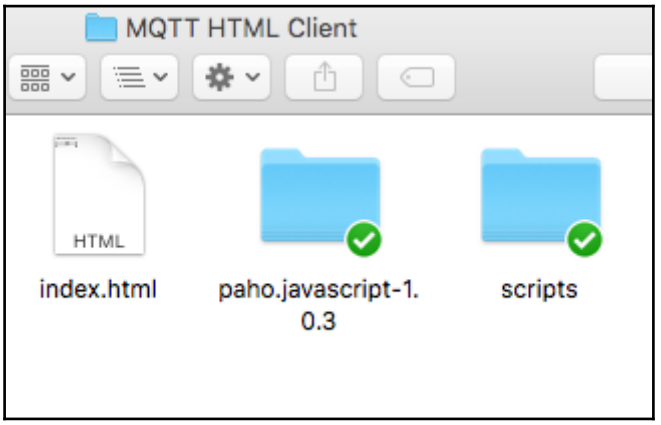

Select a region and data center - Step 2 of 4
Data center
US-East-1 (Northern Virginia)
aws
« Back
Cancel
Confirm


Plan
Cute Cat
Total: $0/month
Name:
T.A.R.A.S
Provider:
Amazon Web Services
Region:
US-East-1 (Northern Virginia)
Tags:

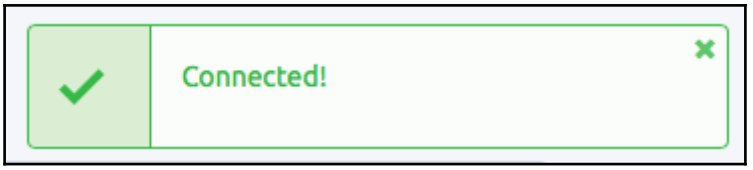

T.A.R.A.S.
Cat
Amazon Web Services US-East-1 (Northern Virginia)
Edit


Instance info
Server
m15.cloudmqtt.com
User
ychzdsuq
Restart
Password
t00uz5DWCsQ_
Rotate
Port
18086
SSL Port
28086
Websockets Port (TLS only)
38086
Connection limit
5


MQTT HTML Client
paho.javascript-1.
0.3
scripts

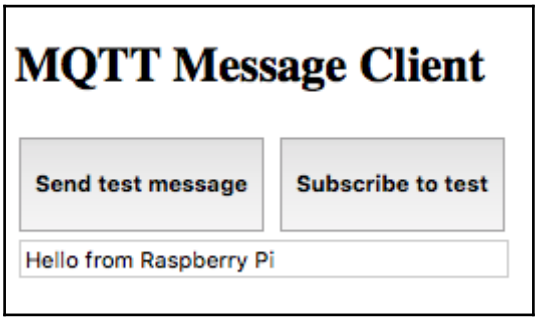

MQTT HTML Client
HTML
index.html
paho.javascript-1.
0.3
scripts


MQTT Message Client
Send test Message
Subscribe to test
Waiting for MQTT message

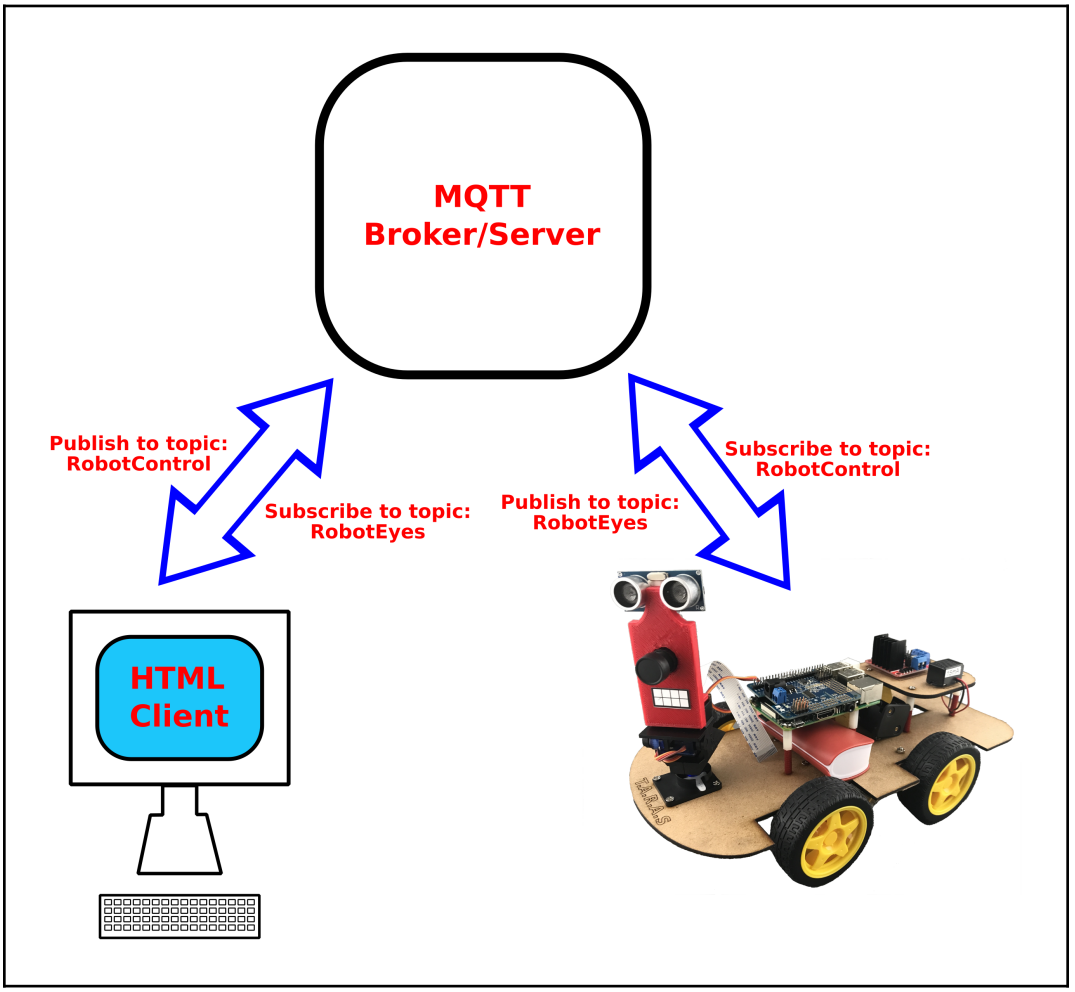

Connected!

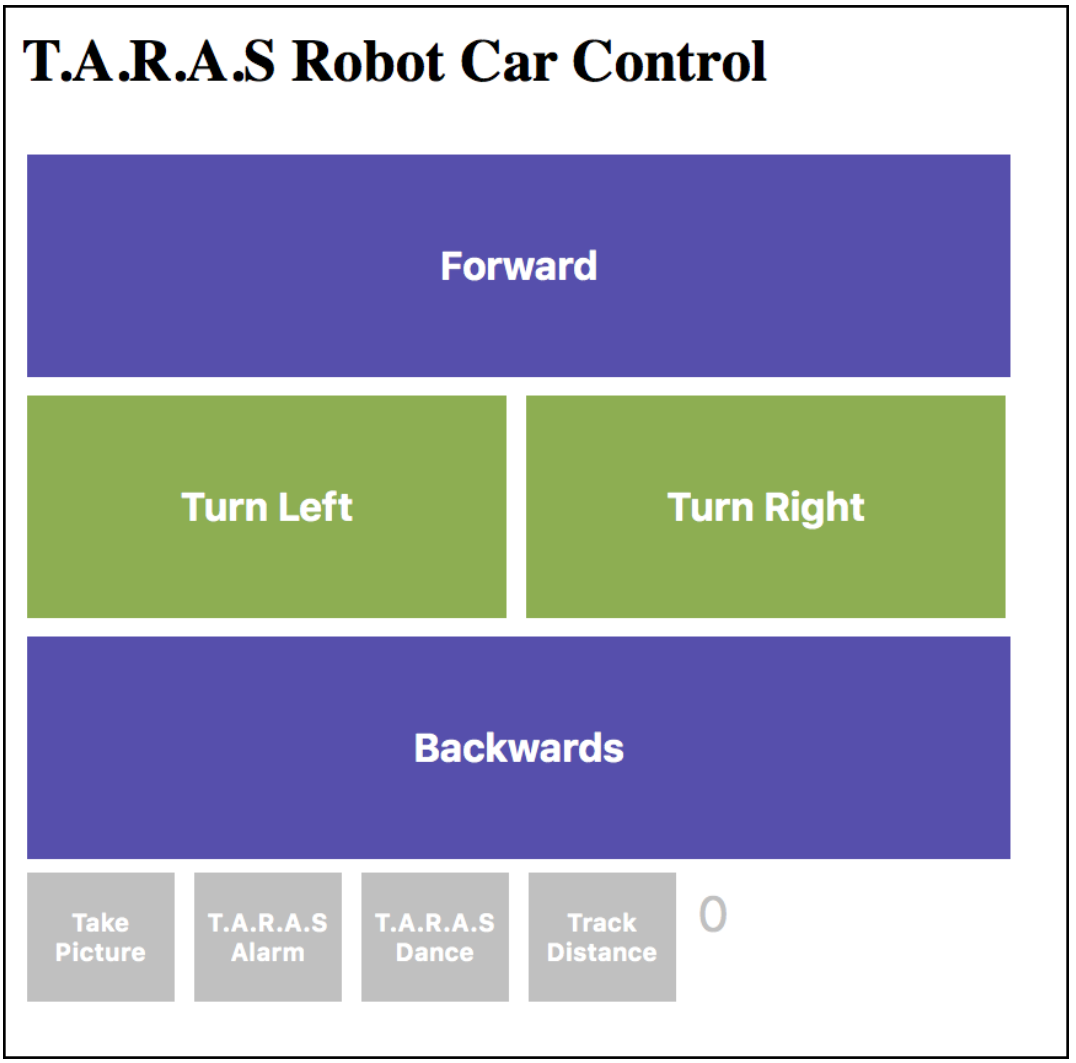

Received messages
Topic
Message
test
Hello from JavaScript client

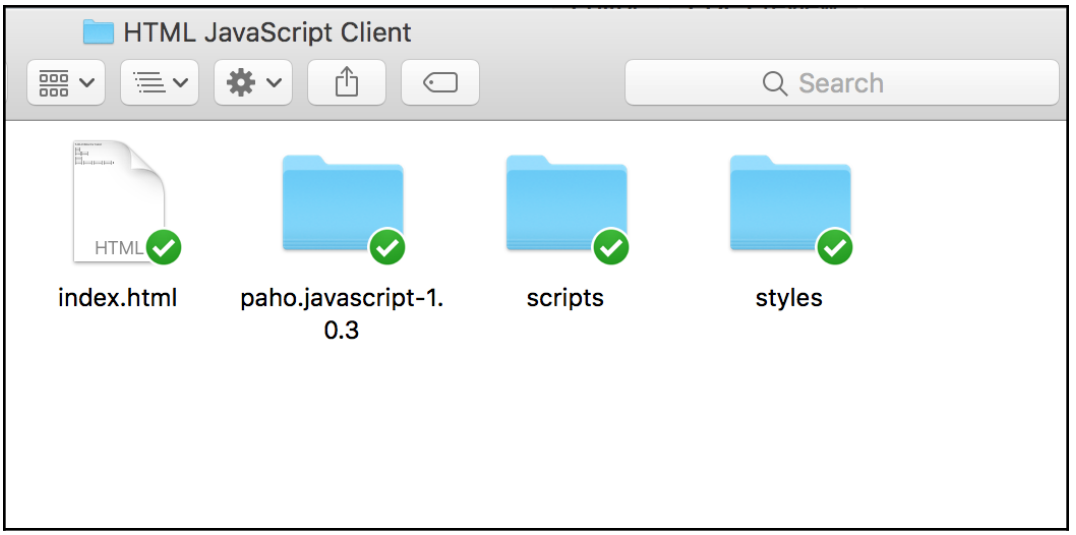

Received messages
Topic
Message
test
Hello from Raspberry Pi

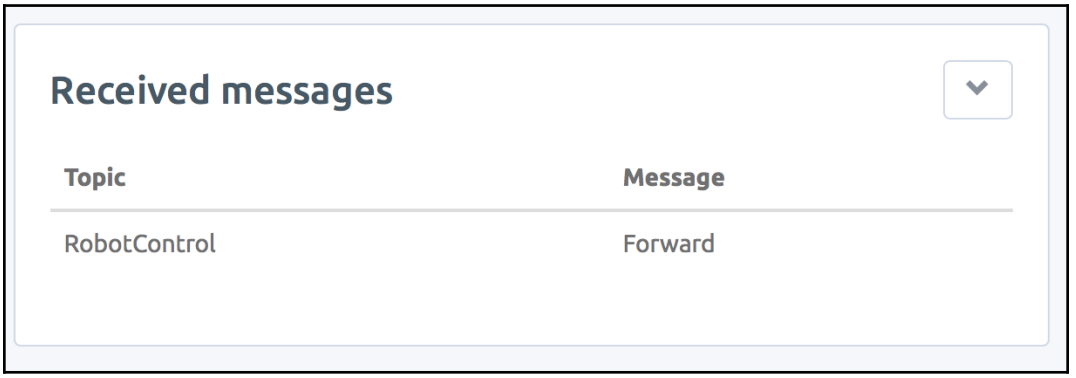

MQTT Message Client
Send test message
Subscribe to test
Hello from Raspberry Pi

Chapter 27: Putting It All Together




T.A.R.A.S Robot Car Control
Forward
Turn Left
Turn Right
Backwards
Take Picture
T.A.R.A.S Alarm
T.A.R.A.S Dance
Track Distance
0


HTML JavaScript Client
Search
HTML
index.html
paho.javascript-1.0.3
scripts
styles

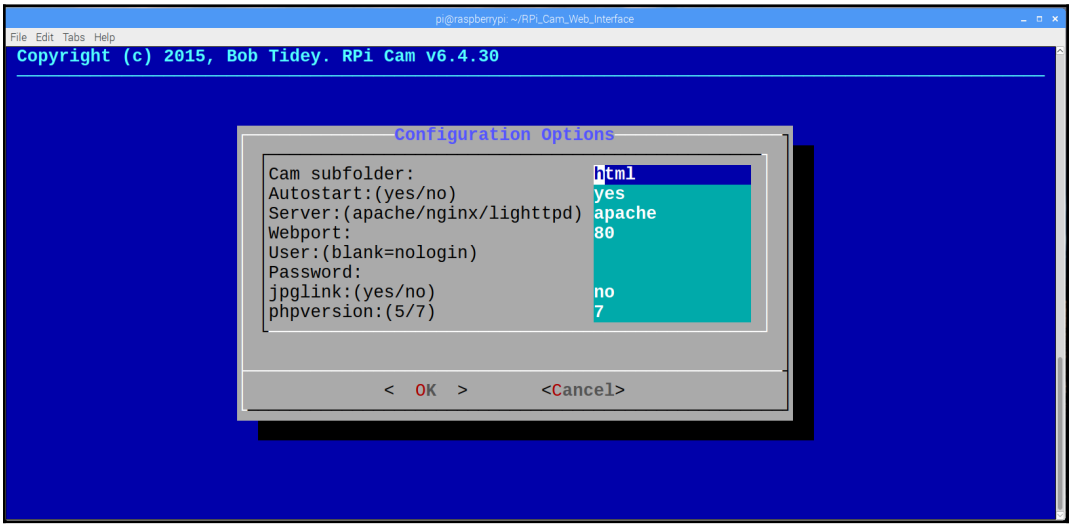

Received messages
Topic
Message
RobotControl
Forward

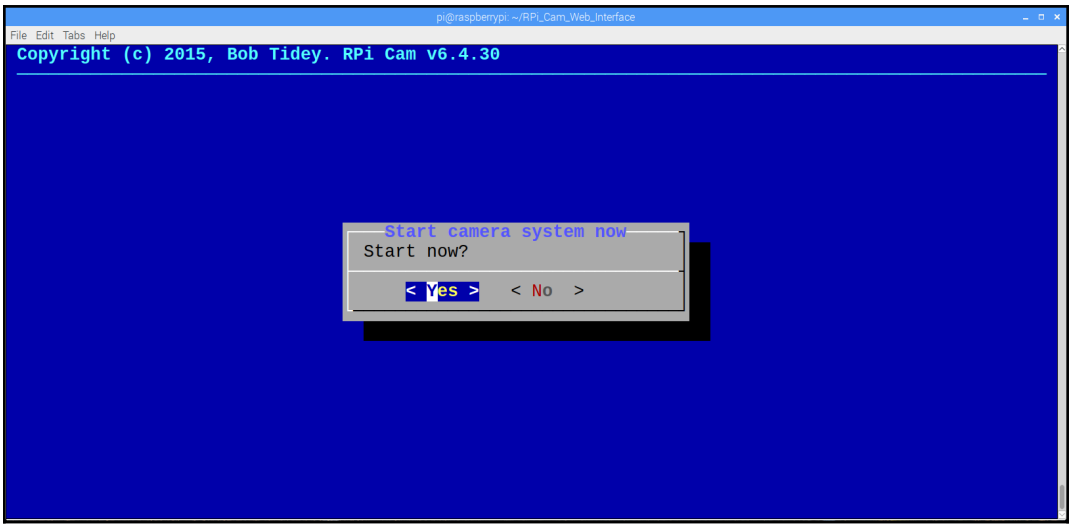

Forward

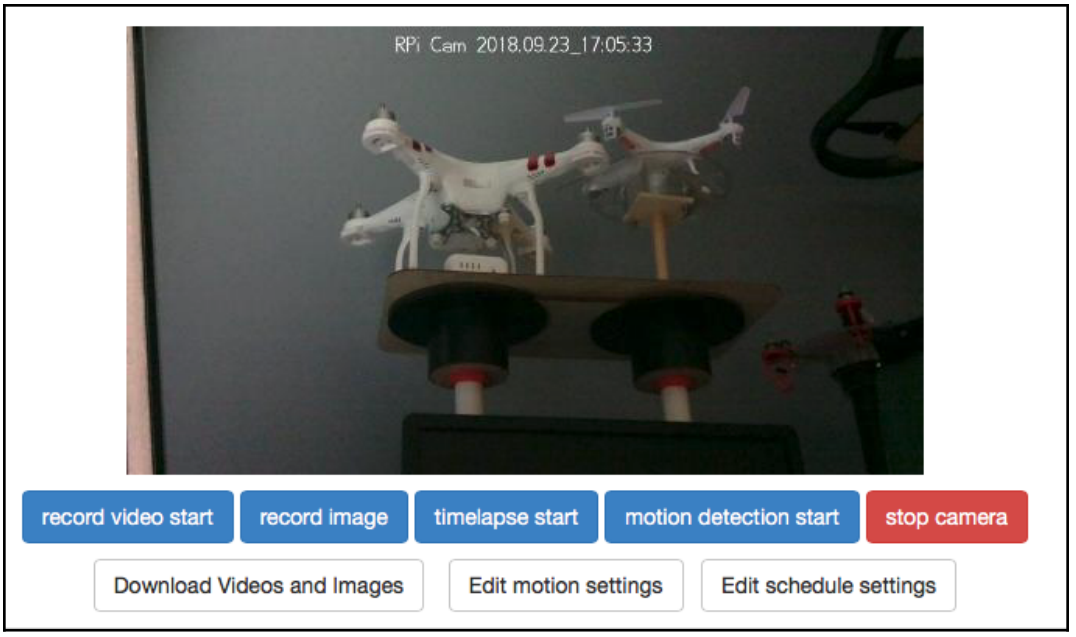

Take Picture
T.A.R.A.S Alarm
T.A.R.A.S Dance


Track Distance


Subscribed to distance data
Close


pi@raspberrypi: ~/RPi_Cam_Web_Interface
File Edit Tabs Help
Copyright (c) 2015, Bob Tidey. RPi Cam v6.4.30
Configuration Options
Cam subfolder: html
Autostart:(yes/no) yes
Server:(apache/nginx/lighttpd) apache
Webport: 80
User:(blank=nologin)
Password:
jpglink:(yes/no) no
phpversion:(5/7) 7
< OK > <Cancel>


pi@raspberrypi: ~/RPi_Cam_Web_Interface
File Edit Tabs Help
Copyright (c) 2015, Bob Tidey. RPi Cam v6.4.30
Start camera system now
Start now?
< Yes >
< No >


RPi Cam 2018.09.23_17:05:33
record video start
record image
timelapse start
motion detection start
stop camera
Download Videos and Images
Edit motion settings
Edit schedule settings


G
H
B

Graphics Bundle Ends Here

Index