ECEN 5823-001 / -001B

Internet of Things Embedded Firmware

Lecture #18

26 October 2017

Be Boulder.

University of Colorado Boulder

Agenda

- Class announcements
- Feedback on the Mid-Term
- Course Project
- Bluetooth Smart

Class Announcements

- BLE Health Temperature Service assignment is due at 11:59pm on Sunday, October 29th, 2017
- The goal is to have the mid-term graded by 12:00pm on Friday the 27th
- Course Project is being assigned

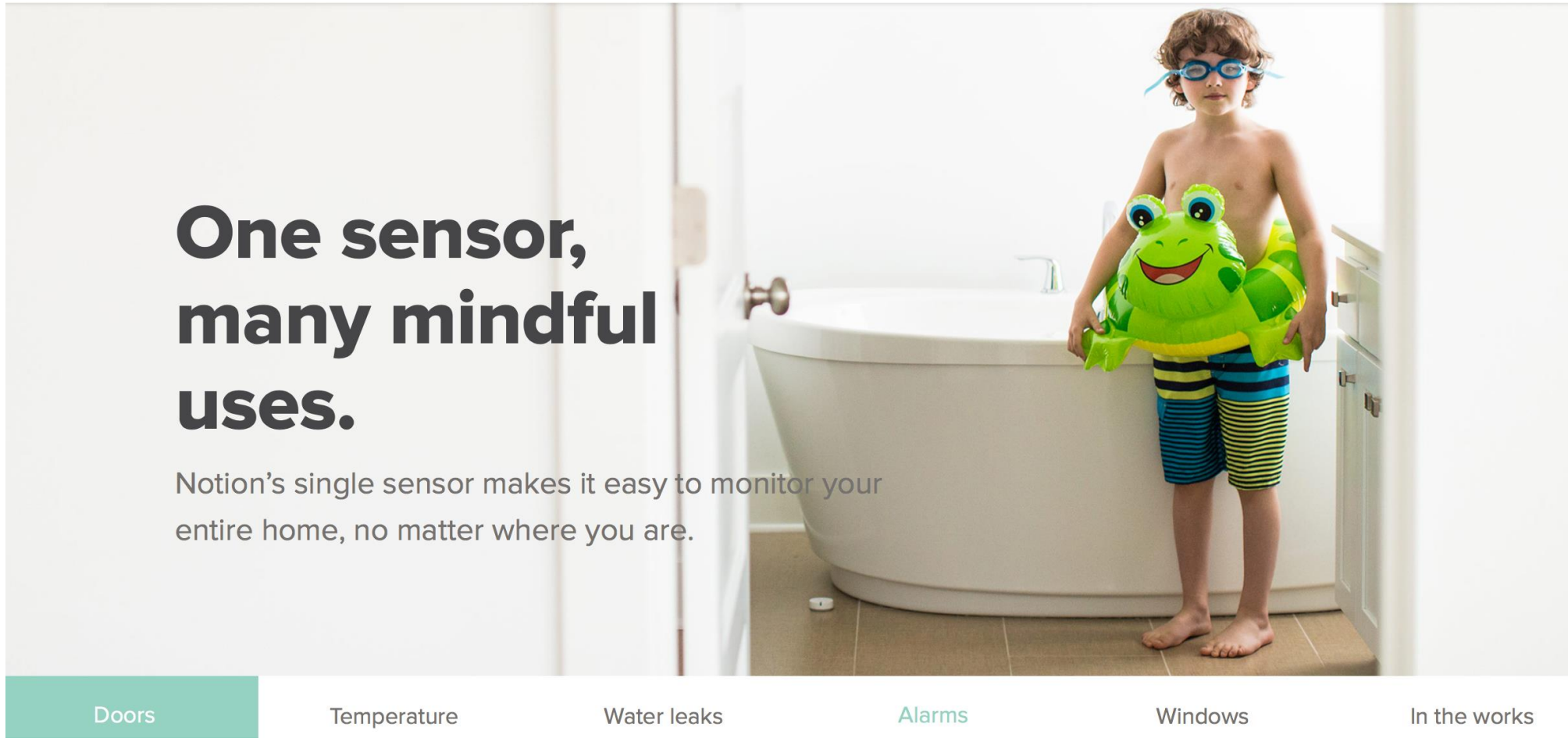
Class Announcements

- The Society of Embedded Engineers is organizing a Tech Talk next week about the V2X technology from Qualcomm. A Senior Staff Engineer/Manager from Qualcomm will get into the details of Cellular V2X (Vehicle to Everything) and there will be some live demonstration!
  - The talk will be presented by:
  - Anshu Boonapalli, Senior Staff Engineer/Manager for the Boulder Office at Qualcomm
  - Location: ITLL 1B50
  - Time: October 30th (Monday), 6:00pm
- Free Pizza!

Mid-Term (Preliminary feedback)

- First observation is that students which did not attend class regularly did poorer on the Mid-Term
- Statistics (Mid-Term plus bonus questions)
  - Top grade: 95%
  - Mean: 80%
- More feedback will be provided on Tuesday the 31st

notion


One sensor, many mindful uses.
Notion's single sensor makes it easy to monitor your entire home, no matter where you are.
Doors
Temperature
Water leaks
Alarms
Windows
In the works

Course Project Assigned

ECEN 5823
Course Project Assignment
Fall 2017

Objective: To further develop your Bluetooth Smart / Low Energy knowledge and experience by exploring an area of interest.

This course project can be done individually or in teams of up to 4 students. The requirement of the project changes depending on the number of people on the project.

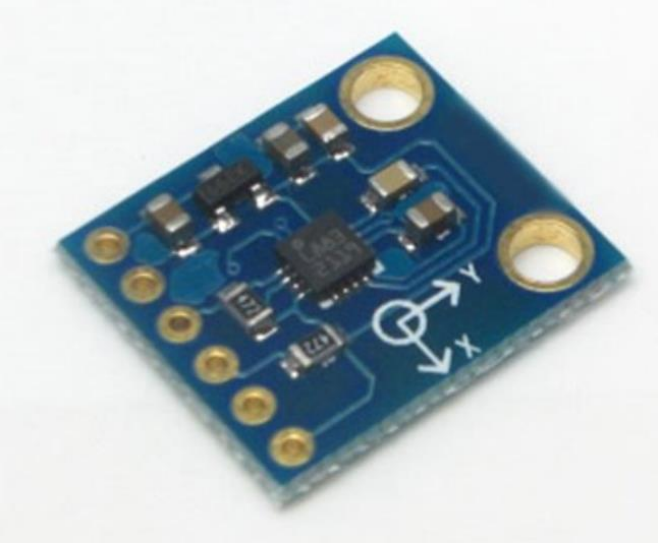
Individually: Can choose the project to be a Bluetooth Client and/or Server
Team of 2: One team member will do a Bluetooth Client that will communicate to the other student's Bluetooth Server.
Team of 3 or 4: Implement a Bluetooth Mesh Network

Course Project credit breakdown (tentative)

- Project proposal — 10.0% of course project grade
- Status report 1 — 10.0%
- Status report 2 — 10.0%
- Final project — 50.0%
  - OTA firmware update — 10.0%
  - Persistence system parameters / data — 10.0%
  - Adaptive TX power — 5.0%
  - SPI LCD display — 10.0%
  - DMA to SPI LDC display — 10.0%
  - Additional sensor — 10.0%
  - 2 new BLE services or profiles — 10.0%
  - BLE security implementation — 5.0%
  - Low Energy Implementation – BG — 20.0%
  - Cohesive application — 5.0%
  - WOW factor (impressive app/demo) — 5.0%
- Project Report — 20.0%
  - Total — 100.0%

Sensor examples

- 3-axis Magnetometer
  - GY-271 HMC5883L Triple Axis Compass Magnetometer Sensor Module



FEATURES	BENEFITS
3-Axis Magnetoresistive Sensors and ASIC in a 3.0x3.0x0.9mm LCC Surface Mount Package	Small Size for Highly Integrated Products. Just Add a Micro-Controller Interface, Plus Two External SMT Capacitors Designed for High Volume, Cost Sensitive OEM Designs Easy to Assemble & Compatible with High Speed SMT Assembly
12-Bit ADC Coupled with Low Noise AMR Sensors Achieves 2 milli-gauss Field Resolution in ±8 Gauss Fields	Enables 1° to 2° Degree Compass Heading Accuracy
Built-In Self Test	Enables Low-Cost Functionality Test after Assembly in Production
Low Voltage Operations (2.16 to 3.6V) and Low Power Consumption (100 µA)	Compatible for Battery Powered Applications
Built-In Strap Drive Circuits	Set/Reset and Offset Strap Drivers for Degaussing, Self Test, and Offset Compensation
I^2C Digital Interface	Popular Two-Wire Serial Data Interface for Consumer Electronics
Lead Free Package Construction	RoHS Compliance
Wide Magnetic Field Range (+/-8 Oe)	Sensors Can Be Used in Strong Magnetic Field Environments with a 1° to 2° Degree Compass Heading Accuracy
Software and Algorithm Support Available	Compassing Heading, Hard Iron, Soft Iron, and Auto Calibration Libraries Available
Fast 160 Hz Maximum Output Rate	Enables Pedestrian Navigation and LBS Applications

Picture from Amazon.com and data from Honeywell's HMC5883L data sheet

Sensor examples

- Barometric Pressure Temperature Sensor
  - BME280 Pressure Temperature Sensor Module with I2C

Digital Pressure Sensor

Key parameters

• Pressure range	300 … 1100 hPa (equiv. to +9000…-500 m above/below sea level)
• Package	8-pin LGA metal-lid Footprint : 2.0 × 2.5 mm², height: 0.95 mm
• Relative accuracy (950 … 1050hPa @25°C)	±0.12 hPa, equiv. to ±1 m
• Absolute accuracy (950 …1050 hPa, 0 …+40 °C)	typ. ±1 hPa
• Temperature coefficient offset (25 … 40°C @900hPa)	1.5 Pa/K, equiv. to 12.6 cm/K
• Digital interfaces	I²C (up to 3.4 MHz) SPI (3 and 4 wire, up to 10 MHz)
• Current consumption	2.7µA @ 1 Hz sampling rate
• Temperature range	-40 … +85 °C
• RoHS compliant, halogen-free	
• MSL 1	

Sensor supply voltage	V_{DD}	ripple max. 50mVpp	1.71	1.8	3.6	V
Interface supply voltage	V_{DDIO}		1.2	1.8	3.6	V

Picture from Amazon and Key parameters from Bosch BME280 data sheet

Sensor examples

- 3-axis accelerometer
  - SparkFun Triple Axis Accelerometer Breakout - MMA8452Q

MMA8452Q

Features

- 1.95V to 3.6V supply voltage
- 1.6V to 3.6V interface voltage
- ±2g/±4g/±8g dynamically selectable full-scale
- Output Data Rates (ODR) from 1.56 Hz to 800 Hz
- 99 µg/√Hz noise
- 12-bit and 8-bit digital output
- I^2C digital output interface
- Two programmable interrupt pins for six interrupt sources
- Three embedded channels of motion detection
  - Freefall or Motion Detection: 1 channel
  - Pulse Detection: 1 channel
  - Transient Detection: 1 channel
    - Orientation (Portrait/Landscape) detection with set hysteresis
    - Automatic ODR change for Auto-WAKE and return to SLEEP
    - High-Pass Filter Data available real-time
    - Self-Test
    - RoHS compliant
    - Current Consumption: 6 µA to 165 µA

Photo from the Sparkfun.com and the MMA8452Q features from its NXP data sheet

Sensor examples

- Gesture sensor
  - Sparkfun

- Features •
  - Ambient Light and RGB Color Sensing - UV and IR blocking filters
  - Proximity Sensing
    - Programmable driver for IR LED current - Saturation indicator bit
  - Complex Gesture Sensing
    - Four separate diodes sensitive to different directions
    - Interrupt driven I2C communication
  - I2C-bus Fast Mode Compatible Interface
    - Data Rates up to 400 kHz
    - Dedicated Interrupt Pin

Parameter	Symbol	Min	Typ	Max	Units	Test Conditions
IDD supply current [1]	I_{DD}		200	250	µA	Active ALS state PON = AEN = 1, PEN = 0
			790			Proximity, LDR pulse ON, PPulse = 8 (I_{LDR} not included)
			790			Gesture, LDR pulse ON, GPulse = 8 (I_{LDR} not included)
			38			Wait state

Place your sensor request on the following Google Sheet !!!

- https://docs.google.com/a/colorado.edu/spreadsheets/d/1fNUuaJ-69DEbhiX6ZufN6sRnKK9lEzBg-cTpXQGtF5o/edit?usp=sharing

Project Proposal Assigned

ECEN 5823
Project Proposal Assignment
Fall 2017

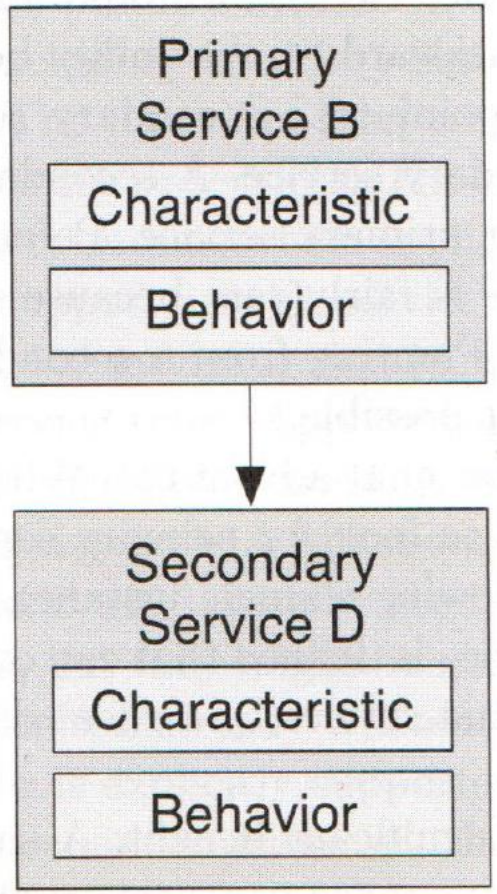
Objective: To define and develop the scope of Course Project in ECEN 4593, Fall 2017.

Note: You can use as much or as little of your previously developed code for ECEN 5823.

Project Proposal Due Date: Saturday, November 4th, at 11:59pm via D2L drop box

BLE: Services

- Primary and secondary services
  - A primary service exposes what a device typically does
    - Example: Battery gas gauge's primary service is to provide how much charge is in the battery
  - A device can have both a primary and secondary services
    - If a devices supports a Service B, Service B would be instantiated as a primary service
    - If the device has additional information, but not associated with the what the device does, this secondary service, Service D, would be instantiated as a secondary service.
  - A secondary service is an encapsulation of behavior and characteristics that are not something that a user would necessarily need to understand
    - Example: The battery gas gauge providing the temperature of the battery



BLE: Services

- Secondary services can only be found by reference and must always have another service that points to them
- A primary service can have more than one secondary service which can create a tree of services
- Primary services can easily be located by a client that is looking for a particular service
  - This enables a simple client to locate whether a desired service is on a device with minimal effort
- Once the client builds up the “tree” of services or multiple “trees,” a “forest,” the client can pass this information to the application to determine how to use the services

BLE: Services

- Service Declaration
  - A service is grouped by using a service declaration which is an attribute type of Primary Service or Secondary Service
  - Simple clients are devices that have no user interface but can still use services on a peer device
    - A simple device can just search for the primary services and find the services that it requires
    - Since the Primary services are directly accessible, the simple client does not need to walk through the “tree” of services to locate it

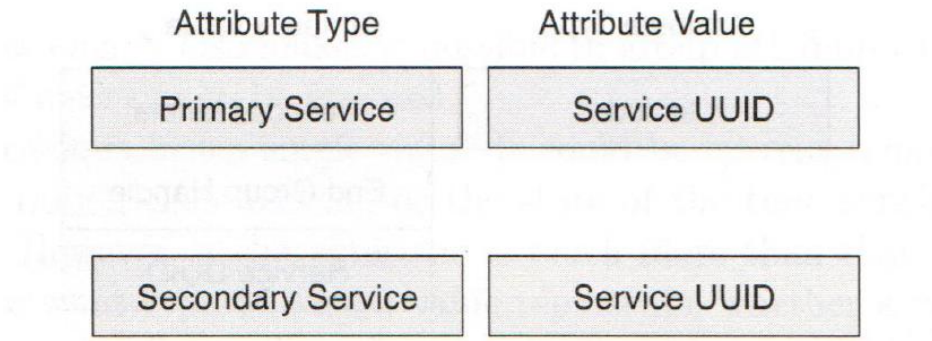


Figure 10–21 Primary and secondary service declaration

BLE: Services

- Including Services
  - Secondary services must be discovered separately
  - Each service can have zero or more Include attributes
  - Include attributes always immediately follow the service declaration and come before any other attributes for the service
  - The Include attribute definitions encompass the handle range for the referenced service along with the Service UUID
    - Enabling quick discovery of the referenced services, their group attributes, and type of services

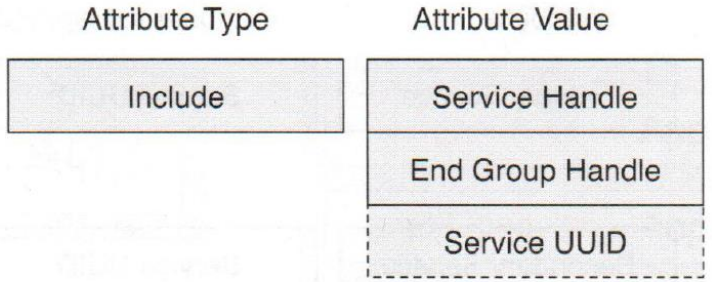


Figure 10–22 The structure of the Include declaration

Handle	Type	Value
0x0001	Primary Service	Service B
0x0002	Include	0x0200, 0x0209, Service D
...	...	...
0x0100	Primary Service	Service A
0x0102	Include	0x0300, 0x0317, Service C
...	...	...
0x0200	Secondary Service	Service D
...	...	...
0x0300	Primary Service	Service C
...	...	...

Figure 10–23 An example of an attribute database of Services A(C), B(D)

BLE: Characteristics

- A characteristic is just a single value
- However, a characteristic needs to expose the following:
  - What type of data a value represents
  - Whether a value can be read or written
  - How to configure the value to be indicated or notified or broadcast
  - And what the value means
- To expose this additional information, a characteristic is composed of three basic elements:
  - Declaration: start of the characteristic and groups all the other attributes for the characteristic
  - Value: the actual value of the characteristic
  - Descriptors: the additional information or configuration of the characteristic

BLE: Characteristics

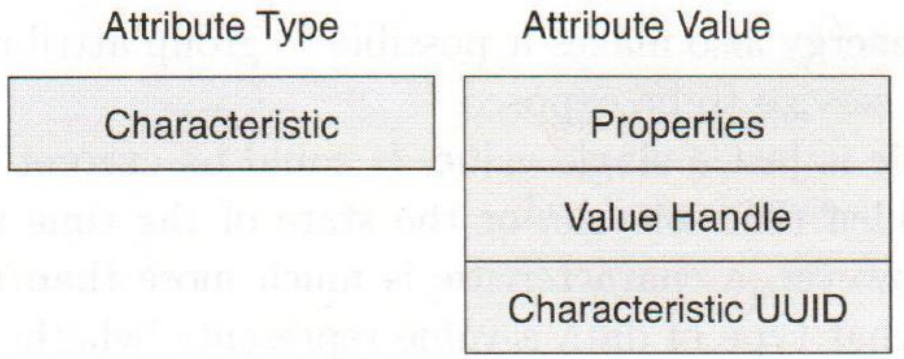


Figure 10–24 Characteristic Declaration

- Characteristic Declaration starts a characteristic and contains three fields:
  - Characteristic properties: specifies if the characteristic value attribute can be read, written, notified, indicated, broadcasted, commanded, or authenticated in a signed write
  - Handle of the value attribute: the handle of the attribute that contains the value for the characteristic
  - Type of characteristic: The characteristic UUID is used to identify the type of characteristic value

BLE: Characteristics

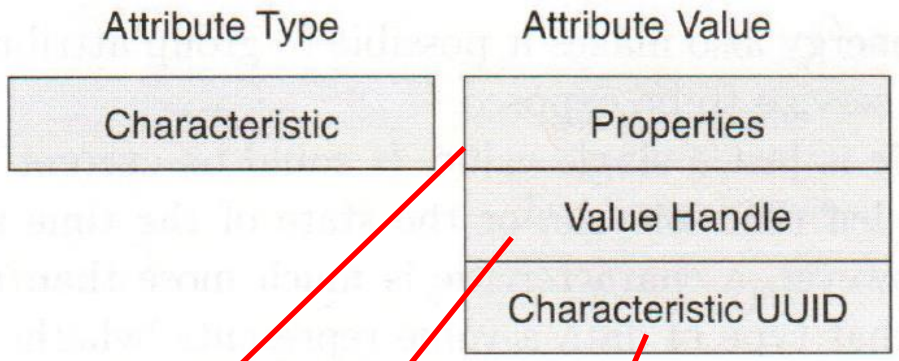


Figure 10–24 Characteristic Declaration

Handle	Type	Value
0x0001	Primary Service	GAP Service
0x0002	Characteristic	read write, 0x0003, Device Name
0x0003	Device Name	"Proximity Tag"
0x0004	Characteristic	read, 0x0005, Appearance
0x0005	Appearance	Tag

Figure 10–25 Characteristic example

BLE: Characteristics

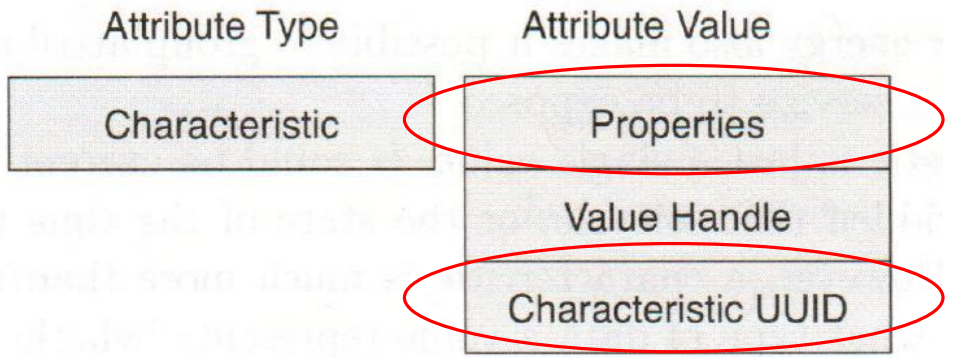
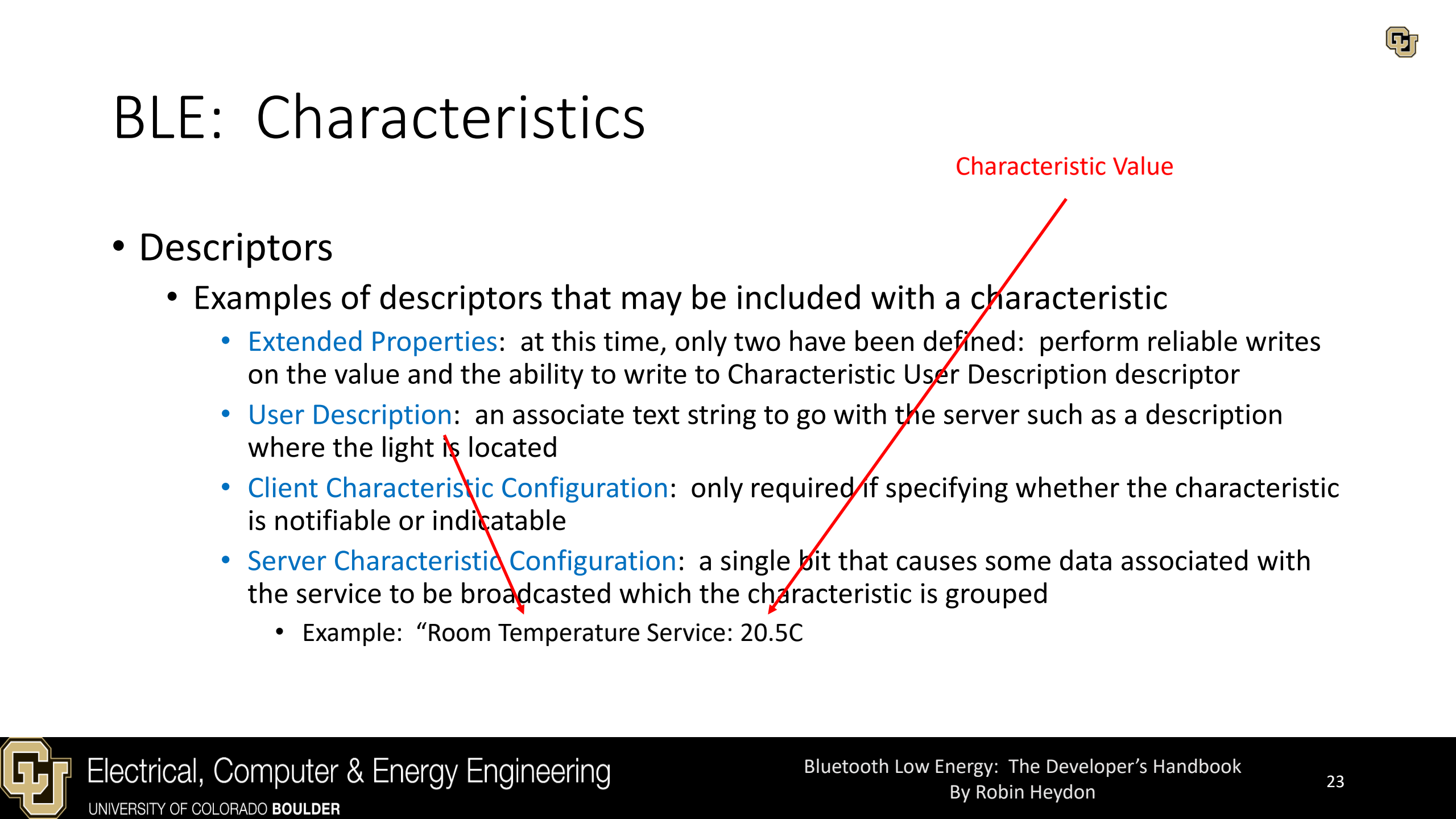


Figure 10–24 Characteristic Declaration

- Characteristic Value
  - The characteristic value is an attribute with the type that must match the characteristic declaration's characteristic UUID field.
  - Types of actions that can be performed on the characteristic value is exposed by the characteristic declaration or in the characteristic extended properties descriptor
  - Characteristics have no behavior, so the service specification with which this characteristic is grouped should specify the behavior exposed by this instance of the characteristic

BLE: Characteristics



- Descriptors
  - Examples of descriptors that may be included with a characteristic
    - Extended Properties: at this time, only two have been defined: perform reliable writes on the value and the ability to write to Characteristic User Description descriptor
    - User Description: an associate text string to go with the server such as a description where the light is located
    - Client Characteristic Configuration: only required if specifying whether the characteristic is notifiable or indicatable
    - Server Characteristic Configuration: a single bit that causes some data associated with the service to be broadcasted which the characteristic is grouped
      - Example: "Room Temperature Service: 20.5C

BLE: Characteristics

Displayed value = characteristic value X $10^{exponent}$

- Descriptors
  - Examples of descriptors that may be included with a characteristic
    - Characteristic Presentation Format: a multiple-field value that contains the following information:
      - Format: similar to variable declarations in C such as int, unsigned_16, float, etc.
      - Exponent: base 10 exponent that is a signed integer
      - Unit: UUID defined in the assigned numbers document such as Temperature in Celsius
      - Namespace: single byte that determines which organization controls the descriptor field
      - Description: a 16-bit unsigned number that is like an adjective. For example, if the thermometer exposes two temperature characteristics, this field could specify whether that particular characteristic is “inside” or “outside”
    - Characteristic Aggregation Format: used to reference two characteristic values that are concatenated together into a “single” value
      - Example: GPS coordinates would have a complex characteristic single value that is actually composed of the latitude and longitude values

BLE: Attribute Protocol

- A simple protocol by which an attribute client can find and access attributes on an attribute server
- The six structured basic operations are:
  - Request
  - Response
  - Command
  - Indication
  - Confirmation
  - Notification

BLE: Attribute Protocol

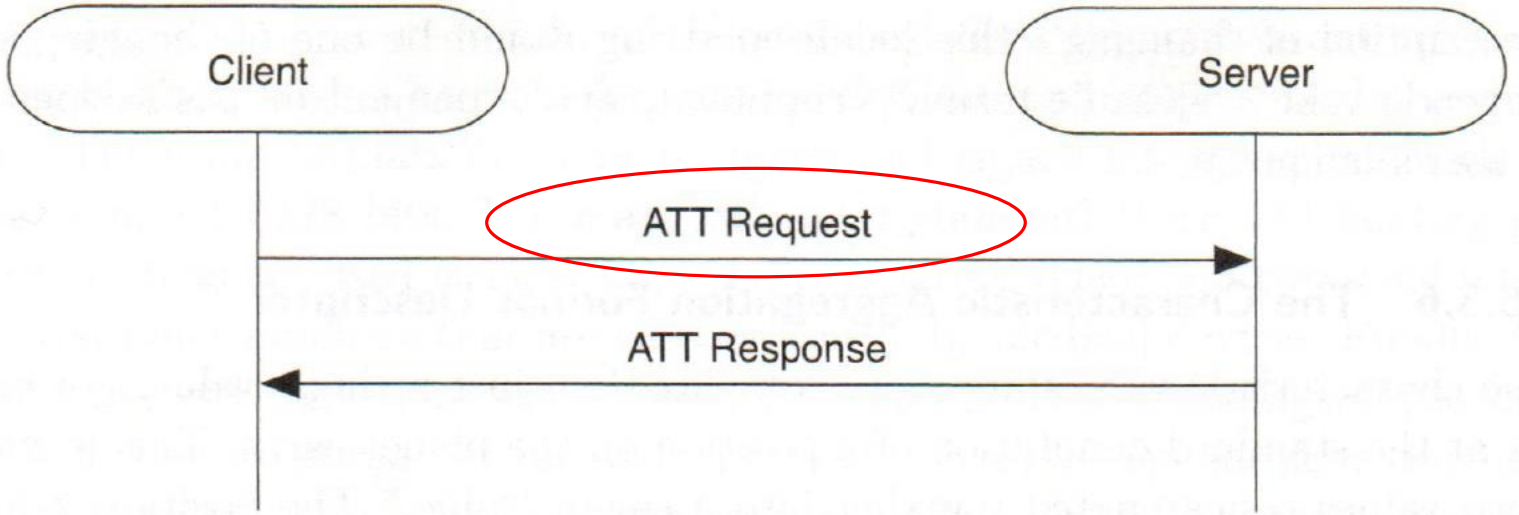


Figure 10–26 An Attribute Protocol request

- The client can send only one request at a time
- The server only has two options for the response
  - A response directly associated with the request
  - An error message

BLE: Attribute Protocol

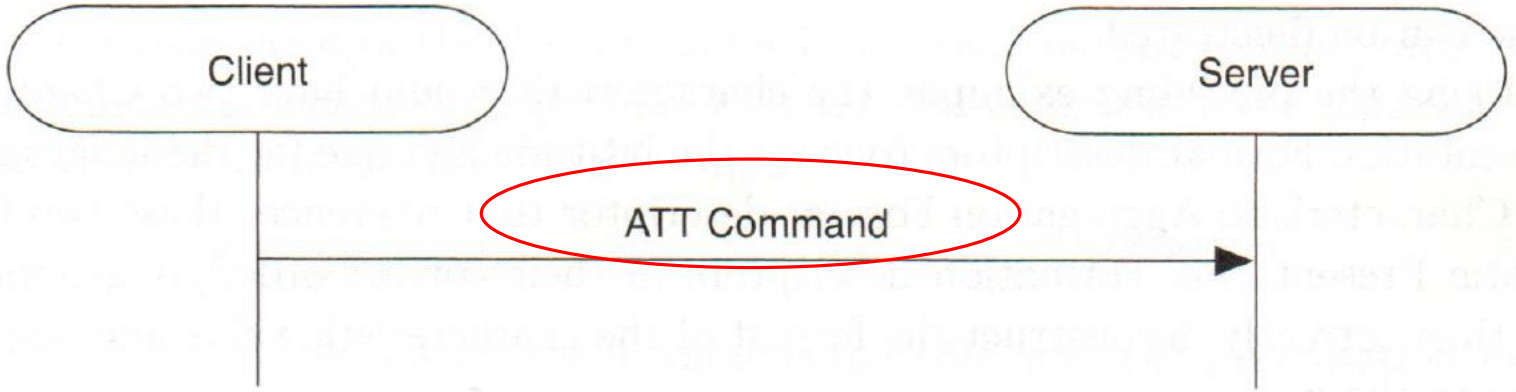


Figure 10–27 An Attribute Protocol command

- The client sends a command to a server but receives no response
- The client sends a command when it wants the server to perform an action with no requirement for an immediate response

BLE: Attribute Protocol

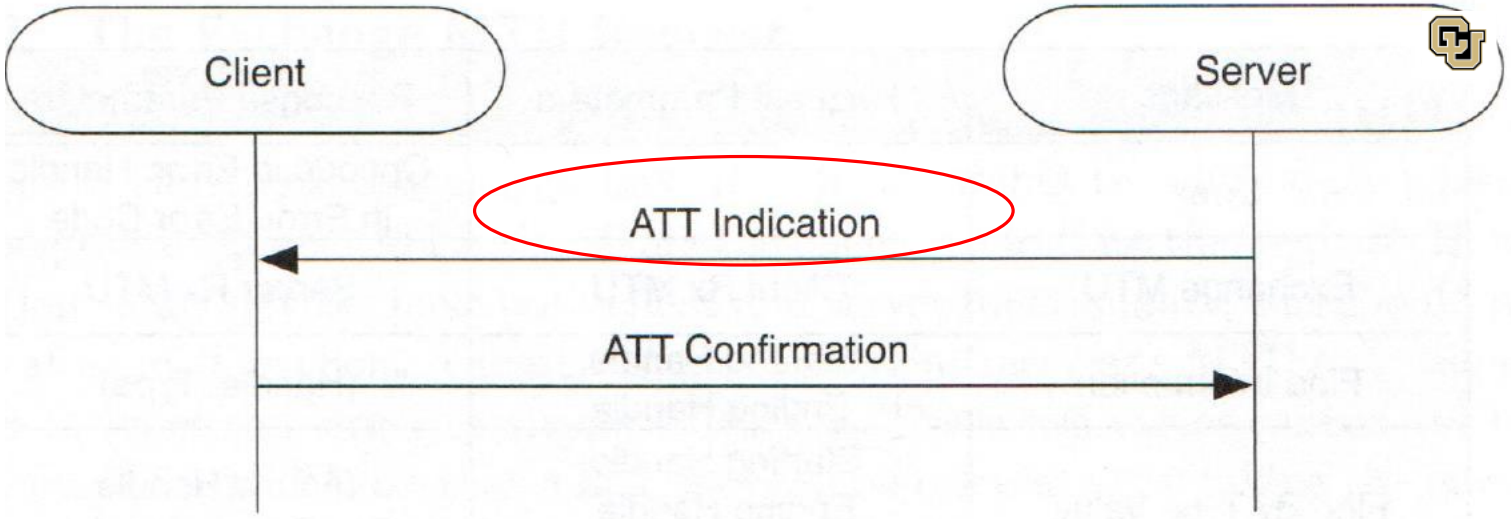


Figure 10–28 An Attribute Protocol indication

- Indications are sent by a server to a client to inform the client that a particular attribute has given a value
- Indications are similar to requests in that only one indication can be sent and a confirmation is required by the client

BLE: Attribute Protocol

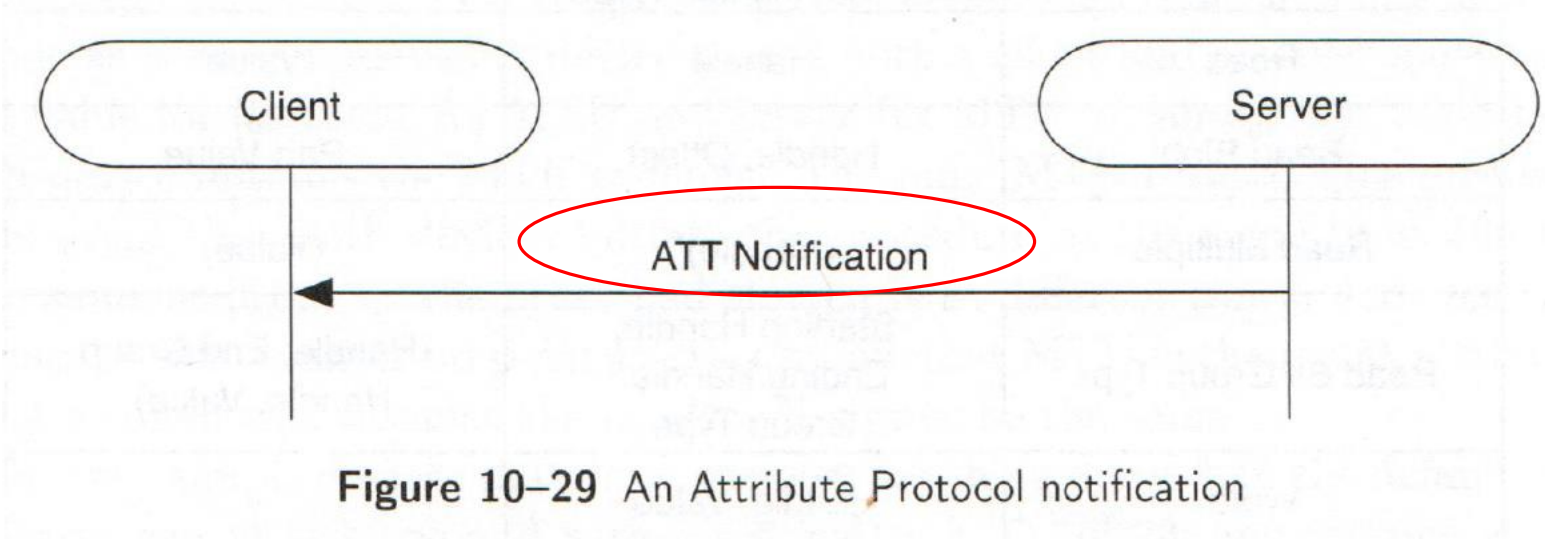


Figure 10–29 An Attribute Protocol notification

- The server can send notification to a client to inform a client that a value of a particular attribute has changed, but does not require a confirmation