# BAP

## Git, GitHub, & GitHub Desktop for beginners

* Create save points that you can go back if you ruin the code.
* GitHub Desktop runs Git.
* Commit = Save point button (can add comment)
* Create brandings of main to then merge the codes (working on one code with 2 can make things harder)

## Bluetooth Versions Explained! (4.0-5.2)

* 4.2 brought
* Bluetooth 4.2 supports IPv6 via Bluetooth Low Energy (BLE), enabling direct internet connectivity without an intermediary, but this typically requires an IPv6-capable **router**. (You guys mentioned no routers between ESP and android, right?)
* Security improvements

ChatGPT:

* **Bluetooth v4.2 includes enhanced encryption** feature: **LE Secure** Connections, which uses **AES-128 encryption** for **securing data in BLE** (Bluetooth Low Energy) communications.
* **In addition** to Bluetooth 4.2 encryption, you can **add end-to-end encryption** by **encrypting the data** on the ESP32 **before transmission** using algorithms like **AES** (symmetric) or **RSA** (asymmetric), ensuring confidentiality from the ESP32 to the server.
* Additionally, if you implement **mutual authentication** (both the ESP32 and the server verifying each other’s identity) and use digital signatures or **hashing** (e.g., HMAC) to verify the integrity of the data, it ensures that the data cannot be manipulated by an attacker during transit.
* **The encryption happens from the ESP32 to the Android phone using Bluetooth Low Energy (BLE) encryption, and then from the Android phone to the server using a secure protocol like HTTPS or TLS, ensuring end-to-end encryption across both connections.**

## End to End Encryption (E2EE) – Computerphile

* There is a communication mechanism in between devices. (Like BLE or WiFi)
* A server is needed in order to pass information from 1 device to the other.
* The channel must be encrypted, or else somebody could get the data before the server does.
* How? Via cryptography.

## 7 Cryptography Concepts EVERY Developer Should Know

* Cryptography takes some data
* Make it impossible to understand data using algorithms
* Only with right credentials you can decrypt the data

1. Hash

Input (any length) goes through Hash function.  
Output of function is a fixed-length value (combo of numbers, letters and symbols).  
Input will always give the same output.  
It is (almost impossible) to get input from output.

1. Salt

Make the password more complex before using salt.

1. Hash-Based Message Authentication Code

Hash that requires a password in order to get the same output when having the same input.

1. Symmetric Encryption

Send a message 🡪 Encrypt the message (called Cipher text) 🡪 Provide key to receiver. (It is random, so it isn’t like hashing where you get the same output)

You create ciphers with different algorithms. (AES256)  
**LIMITATION: SHARED PASSWORD**

1. Keypairs: Public and Private, Public is the shared one, but you have 2 different PRIVATE keys. (RSA, what is used in HTTPS)

ChatGPT:

### How will the android application communicate with the server?

* With HTTPS protocol which uses TLS (Transport Layer Security) to encrypt data during transmission.
* HTTPS with TLS provides protection against (MITM) attacks.   
  The server's certificate is verified by the client (the Android app), preventing an attacker from intercepting or modifying the data without detection.
* It is critical that you use valid SSL certificates and avoid insecure configurations, such as accepting self-signed certificates, to ensure MITM protection.

### What is HTTPS protocol?

TLS uses hashing through cryptographic hash functions (like SHA-256) to ensure data integrity by generating and verifying message digests, which prevent tampering during data transmission by detecting any unauthorized changes.

## Arduino To ESP32: How to Get Started!

* [AZDelivery 3X ESP-32 Dev Kit C V4 compatibel met Arduino inclusief een e-book! : Amazon.com.be: Elektronica](https://www.amazon.com.be/-/nl/AZDelivery-ESP-32-compatibel-Arduino-inclusief/dp/B07Z83H831/ref=sr_1_9?crid=1PLS5UCOXAW4M&dib=eyJ2IjoiMSJ9.VzlPAaPesbbhfNStLTdRjnOjuvpnBQC4JE_mfCyAiEaUZXew9LGheLiuVDmv4xBIXhtVUSUXYP4_WIIw1ACyAZFqMK8QLbw1bcprZpoCjpAFfLw08M2FOs5F8Fpxa77nW2qTfRGwO386Q2K7jFyHE0ZVF6jxCDq64B83S-cXNLSucb3uPiBNLmnYKi618QzreWaeTrY4SGORATsGuPZmaTh7GOSQjeLz_9aLnX_WWBqyhoi5nYk9XKPL3zmTMA555SvSbO3au4PAMki1SVSY-SglmIuRDY4OOu6Z178P3Ds.QUyFYou5yNydn_rtJzBd_r2llT1LeG4HU2NhX30D3Q8&dib_tag=se&keywords=esp%2B32&qid=1728460761&sprefix=esp%2B32%2Caps%2C75&sr=8-9&th=1)
* Desired BLE mode: Access Point 🡪 Send data from sensor (Like a server) to the android app (GATT client), all via Bluetooth. (min 07:00)

ChatGPT guidance:

* You can configure the ESP32 in BLE mode to act as a sensor access point (GATT server) and send data to an Android app (GATT client) via Bluetooth.
* You can transfer data from dual the ESP32 to an Android app by establishing a Bluetooth connection where the ESP32 acts as a GATT server, sending sensor data through characteristics that the Android app, acting as a GATT client, can read.
* Imagine an ESP32 connected to a temperature sensor reading 25°C. The ESP32, acting as a GATT server, creates a "Temperature Service" and a "Temperature Characteristic" with the value 25. The Android app, acting as a GATT client, connects to the ESP32, reads the "Temperature Characteristic," and retrieves the value 25°C from the sensor via BLE. (Servers need defined services that include characteristics)

## **Getting Started with the ESP32 Development Board**

1. Created Github account. (Git repost)
2. Download Git. (Saves history of your code)
3. Downloaded esp-idf (IoT development framework)

Failed, we thought on just using Arduino IDE

## [Apache vs NGINX (youtube.com)](https://www.youtube.com/watch?v=9nyiY-psbMs)

* Both used to create Linux Servers
* Both give Transport Layer Security

ChatGPT: You can use NGINX or Apache to create a Linux server and frameworks like Flask, Django, or Node.js with frontend libraries such as React or Vue.js to design a data dashboard.

## BAP uitgelegd:

* Sensoren beschikbaar die via ESP-32 kan worden aangestuurd.
* Gebruik een android smartphone om sensordata naar centrale server te sturen.
* Doel: data van sensor naar ESP-32, van ESP-32 via BLE naar smartphone, via smartphone naar server.
* Server moet een overzichtelijke dashboard weergeven. (Dashboard is een webapplicatie waarmee gebruikers via een webbrowser overzichtelijke informatie kunnen bekijken)

Regels:

* 3 sensoren (Temp, vocht, lichtintensiteit)

• De microcontroller leest minimaal drie sensoren uit, zoals temperatuur, vochtigheid,

lichtintensiteit, luchtdruk, CO2 of andere relevante sensoren.

• De microcontroller leest de sensoren periodiek uit, bijvoorbeeld één keer per seconde. De

verzamelde sensorgegevens worden gebundeld in datapakketten en via Bluetooth Low

Energy (BLE) naar de smartphone gestuurd. Elk datapakket bevat duidelijk gedefinieerde

velden die uniek corresponderen met de waarde van elke sensor, waardoor de data altijd

eenduidig kan worden geïnterpreteerd.

• De smartphone dient als tussenstation en verzendt de sensordata, ontvangen via Bluetooth

Low Energy, naar een specifieke server. Via de smartphone-applicatie kan de gebruiker

onder andere het serveradres aanpassen en bepalen of de dataoverdracht actief is.

• De server ontvangt de verzonden data en slaat deze in een database. Een overzichtelijk

dashboard visualiseert de opgeslagen gegevens op een duidelijke manier.

• De verzonden data moet strikt vertrouwelijk blijven en de integriteit ervan moet

gegarandeerd zijn. Daarom wordt gebruik gemaakt van encryptie en worden er hashing

toegast om foutieve data te voorkomen.

* 2dehands android kopen?